

電子計算學科의 當面課題

金 淇 龍
(弘益大 電子計算學科)

I

大學에 電子計算學科가 創設된 지 10여 년이 경과하여 많은 電算人材가 사회에 배출된 것은 우리나라의 情報處理産業의 발전을 위해 寄與한 바가 크다. 그러나 최초로 電子計算學科가 설립된 大學(崇田大, 1970년)에 있어서도 卒業生이 社會에 진출한 지는 불과 수년 전으로 이제서야 젊은 층의 책임감이 되어 한창 활약할 단계에 있다. 이러한 단계에서 대학에서의 電算教育과 관련된 제반사항을 집사리 集約하고 評價하기는 어렵다.

電子計算學科(원래는 計算機科學科: Computer Science Department)라는 학과가 대학에 설치된 유래는 '60년대에 先進國, 특히 미국에서의 컴퓨터의 급격한 開發과 活用이 이루어져 우리나라에도 금명간 컴퓨터 利用時代가 도래하고 이에 따르는 專門人材가 필요하다고 豫見하였기 때문이다. 즉 우리나라의 情報處理分野에 있어서 學界, 產業界를 포함한 指導層을 양성하자는 관점에서 광범위하고 高度의 電算技術을 습득한 인재를 공급하고자 한 것이다.

그러나 오늘날의 情報處理産業의 進歩는 너무나 急進의이고 시스템의 규모도 크고 복잡화하여 대학에서는 產業界, 實社會의 最尖端技術을

따라가기 매우 어려운 상황에 직면하고 있는 것이다. 이러한 시기에 大學의 한정된 校修期間과 豫算, 實習施設下에서 가장 效果的인 教育을 실시한다는 것은 電子計算學科의 매우 중요한 當面課題라 생각된다.

II

앞에서도 소개하였듯이 컴퓨터를 專攻하는 學科는 미국의 경우 'Computer Science Department' 하나로 명확히 표시된다. 그러나 우리나라의 경우 너무나 명칭이 다양하고, 인접·유사 학과에서도 원래의 學科名에 電算을 더하여 單一學科가 아닌 복합학과의 인상을 준다. 현재 나와 있는 각 대학의 학과명으로는 電子計算學科, 電子計算機工學科, 電子計算工學科, 電算工學科 등이 제 1군에 속하고, 情報管理工學科, 情報管理學科, 情報工學科, 電算情報工學科가 제 2군에 속하며, 電子電算工學科, 電算統計學科, 計算統計學科 등이 기타에 속한다.

각기 특징이 있다고 하나 일반에게는 內容區分이 가지 않아 학과에 대한 認識度가 問題가 된다. 앞서 표시한 區分에 따르면 제 1군에 속하는 學科의 커리큘럼 內容은 하아드웨어 및 소프트웨어 그리고 시스템 관련 과목이 주로 되어 있다. 하아드웨어 관련 과목으로는 전자계산기

의 구성(organization)과 구조(architecture), 마이크로 및 디지털공학 데이터 통신과 네트워크 등이며, 소프트웨어 관련 과목으로는 프로그래밍言語, 데이터구조, 시스템 프로그래밍, 言語 데이터구조, 오퍼레이팅시스템, 컴파일러, 데이터 베이스, 계산이론, 소프트웨어공학, 정보검색, 시뮬레이션 등이고, 시스템 관련과목으로는 시스템 분석과 설계, 오퍼레이션 리서치 등이 주요 전공과목이다. 제 2군에 있어서는 제 1군의 일부 과목과 경영학, 회계학, 경영통계, 경영정보조직론, 계수관리, 마아케팅, 경영자로서의 등이 위주로 되어 있다. 기타 나머지 구분 學科에서는 제 1군의 일부 과목과 전자공학 과목 또는 통계학에 대한 과목이 주요 과목으로 되어 있다. 여기에 미국의 ACM (Association for Computing Machinery)이 작성, 추천하는 커리큘럼 68과 78을 참고로 기재한다.

〈커리큘럼 68〉

1. 기초과목 : 한 학기당 최소 30시간 이상 교육
 - ① Introduction to Computing
 - ② Computers and Programming
 - ③ Introduction to Discrete Structure
 - ④ Numerical Calculus
2. 중간과목
 - ① Data Structure
 - ② Programming Languages
 - ③ Computer Organization
 - ④ Systems Programming
3. 중간선택과목 : 최소한 두 과목 선택
 - ① Compiler Construction
 - ② Switching Theory
 - ③ Sequential Machine
 - ④ Numerical Analysis I, II
4. 관련 수학과목 : 최소 18시간 이상
 - ① Introductory Calculus
 - ② Mathematical Analysis I
 - ③ Probability
 - ④ Linear Algebra
5. 관련 수학과목 선택 : 최소 2 과목 이상
 - ① Mathematical Analysis II
 - ② Advanced Multivariable Calculus

- ③ Algebraic Structures
- ④ Probability and Statistics

위 커리큘럼은 1968년도에 공표된 것으로 우리나라 초기의 電子計算學科(崇田大)의 족보가 된 바 있다. 각 대학에서도 이를 주로 참작하고 각 대학의 현실, 특히 구성된 교수진에 따라 과목이 좌우되었다. 커리큘럼 68은 그 후 개편되어 커리큘럼 78이 발표되었다. 이 커리큘럼 78은 급속도로 발전하는 컴퓨터 관련 理論과 應用의 擴大에 호응하기 위한 새로운 커리큘럼이었다.

〈커리큘럼 78〉

1. 필수과목
 - ① Computer Programming I
 - ② Computer Programming II
 - ③ Introduction to Computer Systems
 - ④ Introduction to Computer Organization
 - ⑤ Introduction to File Processing
 - ⑥ Operating Systems and Computer Architecture
 - ⑦ Data Structures and Algorithm Analysis
 - ⑧ Organization of Programming Languages
2. Advanced level의 선택과목
 - ① Computer and Society
 - ② Operating System and Computer Architecture II
 - ③ Data Base Management System
 - ④ Artificial Intelligence
 - ⑤ Algorithms
 - ⑥ Software Design and Development
 - ⑦ Theory of Programming Languages
 - ⑧ Automata, Computability and Formal Language
 - ⑨ Numerical Mathematics, Analysis
 - ⑩ Numerical Mathematics, Linear Algebra
3. 관련 수학과목
 - ① Introductory Calculus
 - ② Mathematical Analysis I
 - ③ Probability
 - ④ Linear Algebra

- ⑤ Discrete Structure
 - ⑥ Mathematical Analysis II
 - ⑦ Probability and Statistics
4. Special topics 로 권장하는 과목
- ① Micro Computer Laboratory
 - ② Mini Computer Laboratory
 - ③ Performance Evaluation
 - ④ Telecommunications/Network/Distributed Systems
 - ⑤ System Simulation
 - ⑥ Advanced System Programming
 - ⑦ Graphics
 - ⑧ Compiler Writing Laboratory
 - ⑨ Structured Programming
 - ⑩ Topics in Automata Theory
 - ⑪ Topics in Computability
 - ⑫ Topics in Formal Language Theory
 - ⑬ Simulation and Modeling

위의 내용을 살펴보면 필수과목으로서는 프로그래밍 언어로서 Fortran, Cobol, Assembler, 이론과목으로서 離散구조, 컴퓨터 구조, 데이터 구조, 시스템 프로그래밍, 오퍼레이팅 시스템, 시스템 분석과 설계, 컴파일러, 그리고 관련 수학과목으로서는 선형대수학, 수치계산과 통계학 등이 포함되고 있다. 이외에 선택과목으로는 論理回路, 스위칭理論, 마이크로 컴퓨터, 디지털 시스템, 컴퓨터 알고리즘, 정보이론, OR, 시뮬레이션, 소프트웨어공학, 데이터 베이스, 데이터 통신, MIS, 情報檢索 등이다.

위에서 보는 바와 같이 專攻科目으로서 필수적인 것과 선택과목 등 각 大學마다 실정에 따라 차이가 커버하고 있는 듯하나 문제는 그 내용과 水準일 것이다. 대부분의 學科目的 기준이 되는 텍스트로 原書를 사용하거나 원서를 우리말로 再編한 일부의 텍스트가 있지만 내용과 水準은 大學과정으로서 再檢討가 필요한 것이 많다고 본다. 이것은 역사가 짧은 이 분야의 교수의 대부분이 원리 非專攻分野 또는 인접 분야 出身이거나 大學院과정에서만 이 분야를 공부하여 (평균 3學期間 8개 科目 이수) 교수가 된 것이 대부분이고 대학부터 이 분야를 專攻하고 교수

가 된 경우는 이제 겨우 배출되고 있는 실정이다. 이 분야가 어디까지나 應用學問이고 보면 實地應用과 理論이 병행하여야 하는데 역시 日淺하여 實地를 경험하고 가르치는 교수가 적어 실지와는 거리가 있는 理論에 치중하게 되고 大學院의 科目內容과 水準을 그대로 引用함으로써 大學과정의 종합적인 균형이 이루어지지 못하고 있다. 이러한 問題는 지속적인 硏究檢討와 改善이 필요하겠으나 역시 時間이 지나면 점차 나아지리라 본다.

커리큘럼상 문제가 되는 것은 專攻뿐만 아니라 敎養科目도 문제이다. 대학의 方針에 따라 아직도 專攻에 크게 필요치 않는 敎養科目이 너무 많아 귀중한 시간이 낭비되고 있다. 일일이 그 科目을 지적하지 않더라도 외국 커리큘럼과 비교하면 쉽게 발견할 수 있다. 또한 時時로 발전하는 理論과 應用에 맞추어 科目을 조정할 필요가 있으며 특히 새로운 과목을 과감히 추가하여야 한다. 교수도 讀習하고 배워 가면서 가르치면 되는 것이다. 한 예를 들면 人工智能, 패턴認識, 소프트웨어공학, 그래픽理論과 CAD 등과 같은 이미지處理 분야이다. 특히 이제까지의 情報處理가 대부분 文字情報에 대한 電算處理이었으나 이제는 이미지 情報處理가 급격히 중요시되고 있고 應用범위가 擴大되고 있어서 앞으로 커리큘럼에는 이 분야가 크게 반영되어야 한다고 본다.

III

學問으로 볼 때 情報科學·技術은 지정된 計算機시스템을 사용하여 여러 가지 科學的·技術的 또는 社會的 問題解決의 道具로서 사용하는 user science 的 性格이 강한 면과, ① 計算機에 의한 알고리즘의 設計, 分析, 表現 및 應用 ② 情報의 組織化와 管理에 관한 理解를 목표로 하는 專門的인 報情科學의 면이 있다. 計算機科學은 ①에 重點을 두고 한편으로 하아드웨어에도 관심을 두는 學問이다. 이에 비해 情報學은 情報의 수집, 分析, 組織化와 管理 등에 重點을 두는 학문이다. 여기에서 情報에 관한 학문을 體系化함에 있어 “基礎的 概念이나 方法

論이 存在하는지”라는 문제를 고려해 본다. 이에 대해 一義의이며 명쾌한 답변을 하기란 매우 어려운 일인진 하나 他學問 또는 인접 學問과의 概念 또는 實現方法의 差異를 들어 본다.

計算機科學分野에 있어서는 ‘physical, logical, virtual, computability, recursion, syntax, semantics, reasoning’ 등 다른 科學 또는 工學에서는 경험하지 못한 概念과 이의 實現方法이 있다. 이와 같은 특유의 개념과 技法의 基礎에 깔려 있는 思想, 論理, 技法 등을 구체적 研究對象의 여러 분야, 특히 프로그래밍, 시스템과 프로그래밍 方法論, 알고리즘 解釋, 오퍼레이팅시스템, 컴퓨터 구조, 데이터 베이스, 知識工學과 知識 베이스 등을 통하여 연구하는 것이다.

또한 ‘abstraction, description, verification, refinement, scheduling, documentation’ 등과 같은 概念과 技法은 소프트웨어工學에 必要不可缺하다. 모든 記錄事項을 참조하기 쉽게 작성하는 documentation도 특유한 것 중의 하나이다. 그리고 어느 體系에 있어서 주어진 對象이 어떠한 意味를 갖는지 또는 어떠한 機能을 실현하는지 判定하는 技法, 즉 modeling, analysis, synthesis, evaluation 등도 중요한 것이다.

이상에 열거한 概念과 方法論은 情報科學의 學問體系와 다양하게 연관되어 있어서 그 중 하나 하나를 독립적으로 내세워서 이것만을 對象으로 논할 수는 없는 것이다.

여기서 강조하는 것은 ‘計算機科學’의 固有概念, 技法 및 應用을 敎育과정을 통하여 이수할 수 있는 커리큘럼을 編成하는 것이 電算敎育에 必須的이라는 것을 지적하고자 한다. 이에 대한 하나의 예로서 미국의 Association of Graduate Schools and Council of Graduate Record Examination Board의 下部조직인 Committee of Examiners for the Advanced Computer Science Test가 현재까지 출제한 테스트 問題의 카테고리別 分類와 이에 대한 概念과 技法을 소개한다.

1. Programming Systems and Methodology (40%)
 - A) Programming Languages and their Pro-

cessors(Evaluation of expression, block structure parameter passing and binding, control structures, assemblers, interpreters)

B) Programming Concepts (Iteration, recursion, modularity, abstraction refinement, verification, documentation)

C) Properties of Algorithms (Time and space requirements of programs, especially of common processes such as sorting and searching; correctness of programs)

D) Data Structures(Linear data structures, list structures, strings, stacks, queues, trees)

E) Operating systems (Scheduling, resource and storage allocation, interrupts, synchronization, addressing, techniques, file structures, editors, batch/time sharing, networks/communication)

2. Computer Systems (20%)

A) Logic Design (Switching algebra, combinatorial and Sequential networks)

B) Implementation of Computer Arithmetic (Codes, number representation, add/subtract/multiply/divide)

C) Processor Organization (Instruction sets, registers, data and control flow, storage)

D) System Architecture (Configuration of/ and communication among processors, memories, and I/O devices)

3. Theory of Computation (15%)

A) Automata Theory (Sequential machine, transitions, regular expression, turing machines, nondeterministic finite automata)

B) Analysis of algorithms (Complexity of specific algorithms, exact/asymptotic/ lower bound analysis of time/space complexity, correctness)

C) Formal Language (Regular and context-free grammars/ languages, simple properties such as emptiness or ambiguity)

4. Computational Mathematics (20%)

A) Discrete structures(Logic, sets relations, functions, Boolean algebra, linear algebra,

graph theory, combinatorics)

B) Numerical Mathematics(Arithmetic, number representation, numerical algorithm, error analysis, discrete probability and elementary statistics)

5. Special Topics (5%)

(Simulation and modeling, data management system, information retrieval, artificial intelligence)

다음으로 미국 Bell Lab의 Irving 博士가 주장하는 情報處理技術者에게 요청되는 資質을 열거한다. 이것은 커리큘럼 編成과 教育目標 설정의 指針이 될 것으로 본다.

Qualities looked for (in order of priority) in hiring;

- ① Ability to speak and write clearly
- ② Solid background in mathematics
- ③ Solid foundation in computer science fundamentals
- ④ Problem solving ability
- ⑤ Through knowledge of the software development process
- ⑥ Ability to design the 'people part' of a system
- ⑦ Managerial ability

위의 內容이 반드시 우리의 實情에 비추어 전부 해당되는 것은 아니고 하나의 참고로서 표시한 것이다. 計算機科學은 원래 思考, 判斷, 認識, 學習 등 生物이 실행하고 있는 情報處理의 機構를 해명하고 이 기능을 발휘하는 장치를 人工적으로 만들 때의 기초를 支持하는 것으로서 自然科學뿐만 아니라 人文, 社會科學에 걸친 광범위한 學問分野의 기초를 구성하는 것이다. 이와 같이 관련되는 분야가 매우 廣大하여 이러한 각 분야의 研究成果가 有機적으로 統合됨으로써 비로소 그 成果를 나타낼 수 있다.

여기서 社會 또는 企業에서 바라는 大學에서의 電算教育의 立場이 있다. 大學은 전적으로 社會 또는 企業이 요구하는 實用的 內容만을 교

육할 수는 없고 大學으로서의 균형 있는 專攻科目과 學問的 立場에서 필수적인 內容을 교육하게 된다. 그러나 현실적으로 대학을 졸업하고 企業에 취직하는 것이 至上目標이고 보면 역시 比重은 實用的인 것에 치중하여야 한다고 본다.

일부 大學院으로 진학하는 학생은 좀더 理論的인 것을 추구하여야 된다고 본다. 즉 language 중심 교육에서 data processing 중심의 教育이 필요한 것이다. language 교육에도 high level language 보다는 Assembler language에 대한 교육이 더 중요하다고 본다. data processing 교육을 위해서는 이에 관련된 것으로 시스템 分析과 設計를 비롯하여 實務와 관련된 과목을 교육하여야 하며, 소프트웨어/하드웨어가 보다 더 개발되면 自然言語로 컴퓨터를 이용하게 되므로 여러 가지 language 교육은 앞으로 필요 없게 될 것이다. 따라서 language 자체 교육보다는 問題풀이 能力을 교육시켜야 한다.

이를 위해서는 기본적으로 관련과목을 충분히 教育시키고 高學年에서는 case study나 project를 통해 實技를 익히도록 하여야 한다. 이러한 것을 전부 實習으로 충당할 수 있는데, 문제는 커리큘럼 문제와 대부분의 실무경험 없이 대학원에서만 공부한 교수 문제와 대학의 전산센터의 활용 문제이다.

커리큘럼상으로는 의무적으로 이수해야 할 교양과목과 필수과목이 너무 많고 전공과목에 충당할 수 있는 시간 여유는 적어 실습과목을 충분히 부과할 수가 없다. 제한된 전공과목도 실무경험이 없는 교수가 담당하는 경우 자연히 理論的으로만 하게 되고 現實的인 교육이 안 된다.

컴퓨터센터 문제만 해도 간단하지는 않다. 이제 겨우 각 大學이 수준급의 대형 컴퓨터를 설치하는 하였으나 급증하는 大學 業務處理用으로 전환되고 모니 教育用으로 설치한 것이 業務用으로 되어 버리고 모든 우선권이 부여되고 있다. 계산소 직원도 컴퓨터教育 지원보다는 大學의 각 부서별 업무를 부여받아 이를 전담하게 되어 버려 하나의 사무직원화되고 있다. 따라서 어느 시기에 가서는 大學 업무용 컴퓨터는 그 機種이나 규모를 업무전용으로 정하고 教育 및 研究用은 이에 맞는 機種과 크기를 정하여야 한

다고 본다. 특히 Micro computer의 급속한 발전은 충분히 이러한 問題를 해결해 줄 것으로 본다.

IV

어떤 學問이건 基礎理論이 있고 이를 기반으로 그 학문이 발전하고 應用分野를 지원한다. 日淺한 計算機科學에는 全體를 커버하는 統一的인 基礎理論이 존재하기는 어려운 것으로 보나, 대체적으로 다음의 것이 基礎理論에 속한다고 보면 틀림 없다.

- ① 離散系 數學
- ② 알고리즘의 解析과 設計
- ③ 오토마타와 形式言語
- ④ 計算의 理論

其他

- ① 情報理論
- ② 시스템 理論

計算機科學의 커리큘럼에 基礎理論을 포함시키는 意義는 여러 가지 있는데 그 중 하나는 應用上으로 직접 유용한 技法을 제공한다는 것이다. 數學의 경우를 보아도 이전부터 論理數學은 組合回路論의 中心이었으나 최근에는 프로그램의 정당성의 검증에 채용되었고 소프트웨어工學에 應用하게 되었다. 集合論은 많은 분야에 응용되는데 특히 順序回路와 데이터 構造의 표현 등에 이용된다. 그래프理論도 응용이 커져 決定木, polish notation의 木表現, 스케쥴링의 문제, network내의 flow 문제, 데이터의 구조의 표현 등에 이용된다. 알고리즘의 解析과 設計는 실제로 能率이 좋은 알고리즘 구성에 유용하며, 形式言語 및 順序回路의 理論은 프로그래밍 言語나 컴파일러의 構文記述 및 構文解析에 유용한 것이다.

다음으로 基礎理論은 학생의 直觀力을 양성시키고, 여기서 멀어진 생각을 올바르게 발전시키는 專門的인 判斷力을 연마시킨다. 抽象化와 보편화도 교육상 중요한 도구이다. 학생은 이것에 의해서 知識을 體系化하여 이해할 수가 있다.

일례로서 데이터 構造의 집속관계를 抽象化하여 그래프論적으로 記述할 수 있는 것이다.

하나의 理論은 直接的 應用이 없더라도 知的 基礎를 만들어 줄 수 있다. 여기에 經驗이 추가되면 知的 基盤과 事實에 관한 지식이 一體化되어 미로소 高度의 抽象化와 보편화가 가능하게 되고 洞察力이 우수하게 되어 實行力이 풍부한 技術者의 양성이 가능하게 된다.

이상을 요약하면 計算機科學의 커리큘럼에서 기초이론을 포함시키는 意義를 다음과 같이 요약할 수 있다. 즉 첫째는 理論에서 應用으로의 길을 열어 주는 것, 둘째는 直觀力의 양성과 이의 정당성에 대한 판단력의 함양이며, 셋째는 抽象化 또는 보편화라는 우수한 교육상의 도구로서 이용한다는 것이다. 넷째는 확고한 知的 基礎를 만들어 주는 것이다. 또한 기초이론을 가르치는 목적은 학생들에게 당장 활용할 수 있는 技法을 제공할 뿐만 아니라 長期的으로 學問的 基盤을 주는 데 있는 것이다.

V

컴퓨터 그래픽스는 한때 활발하다가 장기간의 침체를 벗어나 최근 이 분야가 다시 활발해졌다. 이는 우선 가격이 저하하였고, 性能도 향상되었으며, 소프트웨어도 그 方法論이 확립되고 컴퓨터의 이용범위가 확대되어 圖形處理나 表示 및 對話의 機器로서의 必要性이 증대하였기 때문이다. 對話, 즉 interaction이라는 技法은 앞으로 全分野에 걸쳐 활용될 수 있는 것으로서 이에 대한 研究와 시도는 꾸준히 지속되어야 할 것으로 본다. 다음으로 network 技術에 대한 것을 언급하고자 한다.

大學教育에 있어서는 計算機를 하나의 要素라고, 通信技術, 데이터 베이스 技術, man-machine 시스템 技術 등을 종합하여 情報處理시스템을 구축하는 方向으로 교육시켜야 될 것으로 본다. net-working에 대해서는, 學部기간중에는 각개의 技術을 상세히 하는 것보다는 全體로서의 시스템 體系의 개념을 이해시키고, 각 기술의 위치를 표시하는 것이 중요하다. 데이터 베이스 技術에 관해서도 급후 이의 利用方向이 net-

work 를 수반하게 되고 複數의 데이터 베이스와 의 結合이 이루어지는 것을 고려하여야 한다는 것이다.

分散데이터 베이스에 관해서는 많은 어려운 문제가 있다. 複數의 데이터 베이스를 統合한 모델의 확립, 모델간 變換의 技術, copy 에 대한 內容의 一貫性 유지, 障害에 대한 信賴性 유지, 分散資源管理 등이며, 實用的인 技法의 開發이 요망된다. man-machine · interface 에 관한 技

術은 거대한 情報시스템을 활용하는 데 필수적이며, 今후는 人工知能의 技法을 도입하여 知識 베이스를 基반으로 하고, 사용하기 쉬운 高레벨 인터페이스의 構築技術이 중요해질 것으로 본다.

이 외에도 시스템의 性能向上이나 信賴性 확보를 위한 技法 및 性能分析·評價 등의 분야를 敎育에 도입하는 것이 바람직하다고 생각한다. 이상 多角的으로 언급하였으나 지속적 研究가 필요하다. *

〈敎育資料 案內〉

OECD (Organization for Economic Co-operation and Development)와 SRHE(Society for Research into Higher Education)가 발간한 자료 중, 本協議會 資料室이 최근 入手한 고등敎育에 관한 참고 도서는 다음과 같다.

1. *Industry and University: New Forms of Cooperation and Communciation*, OECD: Paris, 1984.
2. *Educational Trends in the 1970s: A Quantative Analysis*, OECD: Paris, 1984.
3. *Reviews of National Policies for Education: U.S.A.*, OECD: Paris, 1981.
4. *School Furniture: Programme on Educational Building*, OECD: Paris, 1981.
5. *The Leverhulme Report: Excellence in Diversity* SRHE: Surrey, 1983.
6. *The Structure and Governance of Higher Education*, SRHE: Surrey, 1983.
7. *Response to Adversity*, SRHE: Surrey, 1983.
8. *Resources and Higher Education*, SRHE: Surrey, 1982.
9. *The Future of Research*, SRHE: Surrey, 1982.
10. *The Arts and Higher Education*, SRHE: Surrey, 1982.
11. *Agenda for Institutional Change in Higher Education*, SRHE: Surrey, 1982.
12. *Higher Education and the Labor Market*, SRHE: Surrey, 1981.
13. *Access to Higher Education*, SRHE: Surrey, 1981.
14. *Training in Curriculum Development and Educational Technology in Higher Ecucation*, SRHE: Surrey, 1980.