

연구 - 교육 인력 양성을 위한 소프트웨어 교과과정

전북대학교 오일석*

전남대학교 박혁로*

1. 서론

연구-교육 분야의 인력이라고 함은 4년제 대학의 소프트웨어 관련 학과를 졸업한 후, 대학원에서 학업을 계속하여 석사학위 혹은 박사학위를 취득하고 각종 연구기관이나 교육기관에서 연구와 교육에 종사하는 사람을 의미한다. 이 글은 이 분야와 관련하여 국내 현황을 살펴보고, 이러한 인력을 길러내기 위한 교육 프로그램을 제안한다. 어느 시대 어느 국가이던지 우수한 인력과 이들이 개발한 새로운 첨단 기술이 국가 발전을 이끄는 엔진 역할을 하여 왔다. 따라서 이 엔진을 만드는 임무를 담당할 인력을 기르기 위한 교육 프로그램의 중요성은 아무리 강조해도 지나치지 않다.

연구-교육 분야의 교육 프로그램은, 산업 현장의 문제를 해결하기 위해 기존 기술을 단순히 적용하는 수준을 넘어 새로운 기술을 연구하는데 필요한 기초 소양을 기를 수 있는 내용을 담고 있어야 한다. 연구는 새로운 개념, 이론, 그리고 기술을 개발하고 이의 유효성과 우수함을 객관적으로 입증하는 것을 기본적인 목적으로 한다. 또한 대학 교육은 고등학교를 갓 졸업한 학생을 4년 동안 교육시켜 우수한 소프트웨어 엔지니어 또는 연구 인력으로 성장시키는 매우 중요한 임무를 띠고 있다. 특히 우리나라의 경우, 학생들이 주로 입시위주의 암기식 교육에 익숙하므로, 이들이 스스로 아이디어를 고안하여 새로운 기술로 발전시키는 창의적인 사고와 소프트웨어 개발 팀 내에서의 공동 작업 수행 능력 등을 갖추도록 교육해야 한다.

연구-교육 분야의 인력이 갖추어야 할 기본 소양은 다음과 같이 세 가지를 들 수 있으며, 적절한 교육

프로그램은 학생들이 이러한 소양을 균형있게 갖추 수 있는 내용이어야 한다. 특히 우리나라의 경우 짧은 산업 사회를 지나 급속히 정보화 사회로 진입함에 따라 여러 사회적인 부작용이 발생하고 있는 만큼, 건강한 정신을 갖도록 세 번째 소양도 소홀히 하면 안될 것이다.

- 1) 수학과 과학의 기초 이론에 대한 이해
- 2) 컴퓨터 과학의 이론과 실제에 대한 이해
- 3) 사회적 임무에 대한 높은 인식

연구-교육 분야의 공부는 특별한 경우를 제외하고는 고등학교에서 이미 어느 정도 이상의 학문에 대한 동기와 학업 성취도를 갖춘 학생이 시작해야 하며, 연구 중심 교육 프로그램은 이러한 학생을 신입생으로 확보할 수 있는 대학교에서 채택해야 무리가 없을 것이다.

현재 국내 대학들은 외부로부터 연구 중심과 교육 중심 중의 하나를 선택하도록 강요받고 있다. 이러한 요구는 캠퍼스 밖에서 발생하고 있으며 시대적인 요구인 만큼 피할 수 없다고 생각된다. 특히 소프트웨어 산업은 짧은 기술 수명과 새로운 응용의 활발한 창출 등으로 그 요구가 다른 학문 분야보다 크다고 하겠다. 하지만 연구와 교육은 밀접한 관계를 가지기 때문에 이러한 요구를 이분법적으로 해석하는 것은 바람직하지 않다고 생각된다. 예를 들어 교육은 충실히 하려면 충분한 양과 질의 교육 조교(TA, Teaching Assistant)가 필요한데, 교육 조교는 대학원생들이 담당하며 연구에 뛰어난 대학원 학생이 임무를 보다 충실히 수행할 것이다. 따라서 연구와 교육 중심 중에 하나를 선택하는 문제가 아니라 무게 중심을 어느 위치에 두느냐 하는 문제로 봐야 할 것이다.

* 종신회원

많은 대학이 두 경우의 사이에 위치하고 있고 대학간에 편차가 크다고 생각된다. 최근의 업적 평가 강화와 SCI 논문에 대한 경쟁 상태 조성 등으로 현재 많은 대학들이 필요 이상으로 무게 중심이 연구 쪽으로 치우쳐 있다는데 이견이 없을 것 같다. 무게 중심을 어디에 두어야 할지는 전적으로 대학교의 여러 상황에 따라 자체적으로 결정할 문제이다. 이와 같이 대학교 차원에서 만들어지는 큰 규모의 결정 이외에도, 교육 프로그램 운영에 관련한 또 다른 중요한 인식과 결정이 필요하다. 어떠한 무게 중심의 위치를 선택하더라도 소속된 대학 교수는 자신의 자질과 성향에 따라 그들도 연구와 교육 중에 무게 중심을 결정할 수 있어야 한다. 대학을 지탱하는 두 축이 교육과 연구이므로 이 과정에서 하나를 더 중요하게 취급하고 나머지를 낮게 취급하면 조화로운 교육 프로그램을 갖출 수 없고 운영 과정에서 심각한 문제가 발생할 것이다.

마지막으로 여기에 제안한 교육 프로그램은 전체 교과 과정의 여러 트랙 중에 하나로 설치하여 연구-교육 분야에 종사하고자 하는 학생들이 자유롭게 선택하도록 운영하는 것이 바람직할 것이다.

2. 인력 현황

국내의 보고서에 의하면 우리나라는 학력 인플레이션 문제를 안고 있다. 특히 박사 배출량은 미국의 유명 대학 규모를 넘고 있다. 다음은 Stanford report에서 인용한 통계이다[1].

소프트웨어 관련 학과 박사 배출 통계(1997-1998년): (박사배출 수/교수 수)

- Stanford 대학교 (22/86)
- Berkeley 대학교 (24/122)
- KAIST (30/29)
- 서울대학교 (12/23)
- 포항공대 (5/17)
- 연세대학교 (5/10)

이 통계에서 알 수 있듯이 현재 국내 대학의 박사 배출량은 과다하다. 따라서 인력 수급 측면에서 공급이 부족하지 않다고 판단된다. 단지, 국내 대학 교원의 직업 시장(job market)이 제대로 형성되어 있지 않아 수요와 공급 법칙이 적용되지 않는 문제가 있다. 대학은 교원 충원을 위한 제원 때문에 한계를 느

끼고 있고, 최근에는 창업 열풍으로 교수에 대한 선호도가 낮아져 지원율도 낮아져 더욱이나 교원 충원에 어려움을 느끼고 있다. 반면 대학 교원을 원하는 박사 취득자는 대학 교원의 문턱이 너무 높다는 인식을 가지고 있다. 따라서 소프트웨어 분야는 우리나라의 전통적인 교수 직업의 특징, 즉 되기는 어렵지만 한번 되면 안주할 수 있는 특징을 타파하는 노력을 통하여 건강한 직업 시장을 형성하는 것이 중요하다 생각된다.

최근 대학 교수 충원에서 나타나는 특징 중의 하나는 산업체 경력자의 임용 비율이 높아지는 것이다. 2001년 국내 하반기 신입 교수 임용 자료에서 보면 총 567명 임용에서 산업체 등 현장 경력 출신자가 79명으로 13.6%를 차지하여 같은 해 상반기 7%에 비해 크게 상승하였다[2]. 이러한 결과는 최근 대학의 실무 교육을 중시하는 정책의 결과로 해석된다. 이러한 현상은 바람직하다고 여겨지며, 특히 소프트웨어 교육의 경우 산업체에서의 경험은 학생들 실습 교육과 지도에 큰 도움이 될 것이다. 특히 소프트웨어 공학과 같은 과목의 경우 산업체에서의 팀워크에 의한 소프트웨어 개발, 검증, 유지보수 등의 경험과 소프트웨어 개발 도구의 사용 경험은 학생 교육에 절대적으로 필요하다 할 수 있다.

소프트웨어 영역의 연구-교육 인력의 수요처인 ETRI와 같은 국가 출연 연구소와 민간 기업 연구소의 인력 수급은 비교적 원활히 이루어지고 있다고 판단된다. 특기할 사항은 IMF 이후 연구원의 벤처 창업이 활발해진 것이다. 이 과정을 통해 연구소에서 개발한 새로운 기술이 산업체에 자연스럽게 이전되고 산업체의 기술 요구가 연구소에 원활히 전달되는 바람직한 human network이 형성되고 있다고 생각된다.

최근 우리나라 전체적으로 대학원 진학률이 감소하고 있다. 특히 2001년의 대학원 신입생의 경우 감소율이 두드러져 새로운 사회 현상으로 해석되기도 한다[3]. 이러한 현상은 현재의 학력 인플레이션을 감안하면 양적으로는 문제가 되지 않고 바람직한 측면도 있으나, 단지 이러한 양의 감소가 질적인 하락과 관련이 있는 경우에는 향후 큰 문제가 될 것이다. 따라서 과학적인 분석을 통해 질적 하락의 정도를 측정하고 문제가 있다고 판단되는 경우 시급히 대책을 세워야 할 것이다.

3. 교육 프로그램의 특성과 요구 사항

연구와 교육에 종사하기 위해서는 산업체의 소프트웨어 개발 엔지니어와는 다른 특수한 능력이 필요할 것이다. 물론 연구와 교육에 종사하는 인력도 풍부한 소프트웨어 개발 경험을 갖는 것이 바람직하다. 특히 교육 분야에 종사하는 인력의 경우 우수한 소프트웨어 기술 인력을 양성해서 국가의 산업 발전에 이바지할 의무를 가지므로 풍부한 소프트웨어 개발 경험을 갖추어야 할 것이다. 한편 연구에 종사하는 인력도 자신의 아이디어를 구현하여 실험하기 위해 소프트웨어 개발 능력이 필요하다. 그러나 이 분야의 인력은 앞으로의 학위 과정에서 연구를 수행하고 그 결과를 논문으로 작성하여 발표하여야 하기 때문에 자신의 논지를 체계적으로 설득할 수 있는 이론적 배경 지식이 소프트웨어 엔지니어보다 더 필요하다고 하겠다. 즉 이들 인력을 양성하기 위해서는 기술적인 세세함보다는 컴퓨터 과학의 원리와 이론적 측면이 보다 강조되어야 할 것으로 생각된다.

Meyer는 최근 IEEE Computer에 기고한 글에서 소프트웨어 교과과정의 목적으로 다음 다섯 가지를 나열하였다[4]. 이러한 틀에서 고려한다면 교육 프로그램은 다섯 가지 모두를 충실하게 달성할 수 있도록 작성되어야 하는데, 다른 분야와 달리 연구-교육 분야의 프로그램은 원리와 수학에 보다 많은 비중을 두어야 한다.

Principles(원리): 전체 영역을 지배하는 변하지 않는 개념

Practices(실행): 우수한 전문인이 보편적으로 사용하는 문제 해결 기법

Applications(응용): 원리와 실행이 적절하게 적용되는 전문 영역

Tools(도구): 원리와 실행의 응용을 도와주는 최신 제품

Mathematics(수학): 많은 현상의 해석을 가능하게 하는 형식적 토대

IEEE와 ACM은 공동의 task force 팀을 구성하여 커리큘럼 2001(Computing Curricula 2001; CC2001)을 개발하였다[5]. CC2001은 컴퓨터 과학의 학부 수준 프로그램에서 교육해야 할 기본으로 다음과 같은 14개 교과 영역을 정의하였다. 연구-교육 분야의 교육 프로그램은 어느 영역도 소홀히 하지 않고 균형있

게 섭렵할 필요가 있다. 특히 SP 영역은 서론에서 언급한 세 번째 소양(사회적 임무에 대한 높은 인식)과 관련하여 다루어야 한다.

이산구조(Discrete Structures : DS)

기초 프로그래밍(Programming Fundamentals: PF)

알고리즘 및 복잡도(Algorithms and Complexity : AL)

프로그래밍 언어(Programming Languages : PL)

컴퓨터 구성 및 구조(Architecture and Organization : AR)

운영체제(Operating System : OS)

네트워크 중심 컴퓨팅(Net-Centric Computing : NC)

인간-기계 상호작용(Human-Computer Interaction : HC)

그래픽스 및 비주얼 컴퓨팅(Graphics and Visual Computing : GV)

지능 시스템(Intelligent System : IS)

정보 관리(Information Management : IM)

소프트웨어 공학(Software Engineering : SE)

사회적, 전문적 이슈(Social and Professional Issues : SP)

컴퓨터 과학 및 수치 해석(Computational Science and Numerical Methods : CN)

CC2001에서는 컴퓨터 과학의 졸업생이 갖추어야 할, 컴퓨터 과학의 이론과 실제에 대한 요건과 그 밖의 일반적인 요건을 제시하고 있다. 그 중 일반적인 요건에 대해 살펴보면 다음과 같다. 모든 분야의 교육 프로그램이 이러한 요건을 중요시해야 하지만 특히 연구-교육 분야의 프로그램은 이러한 일반 요건을 보다 중요하게 다루어야 한다.

Mathematics: 기초 수학

Scientific: 과학적 방법

Familiarity with applications: 응용에 대한 익숙함

Communication skills: 대화 기술(기술 작문, 구두 발표, 건설적 비판의 이해와 제시 등)

Working in teams: 팀워크

4. 특성화된 교과 과정 조사

연구-교육 인력을 양성하기 위해 특성화된 교과 과정으로 Carnegie Mellon 대학 (CMU)의 컴퓨터 과학과를 조사하였다.

4.1 CMU의 Computer Science Department

이 학과의 교과과정은 컴퓨터 과학 분야의 핵심에 대한 학습과 더불어 의무화된 부전공 이수를 통한 이차관심 분야에 대한 깊이 있는 교육을 결합한 형태이다[7]. 또한 이 교과과정은 학생들에게 과학 및 인문학에 대한 폭넓은 선택 기회를 제공하고 있다. 컴퓨터 과학 분야는 여러 다른 학문 분야와 밀접한 관련을 가지는 분야이기 때문에 이렇게 함으로써 학생들은 자신의 관심에 따라 다양한 교육과정을 선택할 수 있도록 하고 있다.

CMU 교과과정의 특징으로는 다음을 들 수 있다.

- (1) 프로그래밍 기술만을 교육하지 않고, 수학과 통계학 분야를 도입함으로써 비록 기술 및 시스템에 대한 변화가 발생해도 여전히 적용할 수 있는 형식적 개념과 도구들을 교육한다.

표 1 CMU의 컴퓨터 과학 영역 교과목

분야	교과 과목
필수	Intermediate/Advanced Programming
필수	Systems Skills in C
필수	Fundamental Data Structures and Algorithms
필수	Principles of Programming
필수	Introduction to Computer Systems
필수	Great Theoretical Ideas in Computer Science
필수	Algorithm Design and Analysis
택일 (응용분야)	AI: Representation and Problem Solving Robotic Manipulation Computer Vision Database Applications Computer Graphics I Mobile Robot Programming Laboratory
택일 (알고리즘 분야)	More Great Theoretical Ideas in Computer Science Symbolic Algebra Combinatorial Analysis Algebraic Structures
택일 (프로그래밍 분야)	Programming Language Design and Processing Constructive Logic Formal Languages and Automata Models of Software Systems Basic Logic Logic and Computation Computability and Incompleteness
택일 (시스템 프로그래밍 분야)	Operating System Design and Implementation Computer Networks

- (2) 집중적인 프로젝트 기반 과정을 운영함으로써 학생들에게 소프트웨어 개발 및 유지 과정에서 발생하는 실질적인 문제점을 숙지하도록 한다.

이 교과과정은 컴퓨터 과학, 수학/통계학, 공학 및 자연과학, 컴퓨터 기술 및 워크샵, 인문학 및 예술, 부전공 등 6개의 영역으로 구성된다. 컴퓨터 과학 영역과 수학/통계학 영역을 중심으로 살펴보면 다음과 같다. 먼저 컴퓨터 과학 영역은 8개의 필수 과목 및 4개의 택일 과목, 그리고 2개의 선택 과목으로 구성된다. 다음으로, 4개의 택일 및 해당 분야 과목들을 정의하고 각 분야마다 1개 이상의 과목을 수강하도록 하고 있다. 수학/통계학 영역은 해석학 등 3개의 필수 과목 및 확률 및 random 프로세스에 대한 1개의 택일 과목으로 구성되어 있다.

표 2 CMU의 수학/통계학 영역 교과목

분야	교과 과목
필수	해석학(Differential Calculus, Integral Calculus, Integration and Differential Equations) Concepts of Mathematics Matrix Algebra (or Linear Algebra)
택일	확률이론 및 random 프로세스, 확률 및 수학적 통계 1

4.2 국내 상황

우리나라에 소프트웨어 관련 학과는 1970년대 말부터 설치되기 시작하였다. 많은 대학에 컴퓨터 과학과(Department of Computer Science)와 컴퓨터 공학과(Department of Computer Engineering)가 설치되어 운영되어 오다가 최근 학부제에 따라 하나로 통합되고 있는 추세이다. 이들 두 학과는 구별되는 특징이 거의 없이 유사한 교과 과정을 유지하여 오다가 학부 통합에 따라 하나의 교과과정으로 통합하고 있다.

표 3은 최근 학부 통합이 이루어진 서울대학교 컴퓨터공학부의 교과과정이다[6]. 이 교과과정은 1980년대의 틀과 내용을 크게 벗어나지 않고 있다. 특수한 경우를 제외하고 현재 우리나라의 소프트웨어 관련 학과 대부분이 이와 유사한 교과과정을 운영하고 있다. 사실 이러한 상황에 대한 인식과 발전적 동기가 이번 특집호를 발간하게 된 직접적인 원인이라고 볼 수 있다.

표 3 서울대학교 컴퓨터공학부 교과과정

학년	1학기	2학기
1학년	컴퓨터개론	이산수학 컴퓨터프로그래밍
2학년	공학수학1 논리설계 논리설계실험 회로이론 선형대수학	공학수학2 시스템프로그래밍 자료구조 전기전자회로실험 전자회로
3학년	오토마타이론 컴퓨터구조 컴퓨터시스템설계 화일처리 세미나 공대공동교과목	데이터베이스 데이터통신 마이크로프로세서응용 스위칭이론 운영체제 프로그래밍언어 프로젝트1
4학년	VLSI회로 알고리즘 컴파일러 컴퓨터네트워크 프로젝트2	소프트웨어공학 소프트웨어응용 신형컴퓨터시스템 인공지능 컴퓨터그래픽스 컴퓨터모델링

5. 연구-교육 인력 양성을 위한 교과 과정

교과과정을 제시하기 전에 현재 우리나라 대학이 처한 상황을 살펴보자. 현재 대부분 대학은 그 동안 견고하게 유지돼오던 학과간 그리고 단과 대학간의 벽을 허물고 학생들에게 보다 폭넓은 전공 선택의 기회를 제공하기 위하여 학부제가 확대되고 있다. 이에 따라 전공 필수 과목의 수에 대한 제약이 엄격해지고 있으며, 최소전공 학점제의 도입으로 대부분의 대학이 36~40학점 정도의 전공 과목만 이수하면 해당 전공 분야의 학사 자격을 수여하고 있다. 따라서 가장 중요한 과목들만 전공 필수로 지정하고 나머지는 학생들이 자유롭게 선택하게 하도록 해야한다. 또한 이러한 상황을 발전적인 방향으로 유도하는 좋은 방안 중의 하나는 교과과정에 여러 트랙을 설치하여 학생들이 자신의 판단에 따라 특정 트랙을 심도있게 공부하도록 하는 것이다.

어떤 과목을 전공 필수로 지정하는가는 해당 교과 과정이 추구하는 목표에 따라서 달라진다. 이 글은 연구-교육 인력 양성에 초점을 맞추고 있으므로, 교육과정의 편성에 있어서 중요 원칙은 프로그래밍 기

술(skill) 혹은 시스템 사용법 교육보다는 문제 해결을 위한 기본적인 사고 능력과 개념적 도구들을 습득하고 이를 응용에 효과적으로 활용하는 훈련에 더 많은 비중을 두어야 한다. 즉 아래에서 인용한 Mary Show의 말처럼, 보일러의 구조를 교육하기보다는 열역학을 교육하는 것이 더 바람직하다는 가정에서 교과과정을 작성하였다.

Let us organize our courses around ideas rather than artifacts. This helps make the objective of the course clear to both the student and faculty. Engineering schools don't teach boiler design — they teach thermodynamics[5].

Mary Show 1992

서론에서 언급했던 연구-교육 인력의 소양, Meyer의 소프트웨어 교과 과정의 목적, 그리고 CMU에서 운영하고 있는 교과과정 특징 등에 나타난 공통적 기저를 바탕으로 다음과 같이 8개 영역으로 분류하였다.

- (1) 수학(Mathematics)
 - Probability and Statistics
 - Calculus
 - Linear Algebra
 - Continuous Math
 - Discrete Math
- (2) 원리(Principles)
 - Automata Theory
 - Algorithm Design and Analysis
 - Computational Complexity
 - Programming Principles and Methodologies
- (3) 도구(Tools)
 - 각 과목에서 적합한 도구 활용
- (4) 실행(Practices)
 - Data Structure
 - Programming Languages
 - Software Engineering
 - Artificial Intelligence
 - Numerical Analysis
- (5) 응용(Applications)
 - 컴퓨터 시스템 분야
 - System Programming
 - Operating System
 - Compiler

- 정보처리 분야
 - Database
 - Bio-informatics
 - Scientific computing
 - Multi-media Processing
 - Computer Graphics
- 컴퓨터 네트워크 분야
 - Computer Networks
 - Networking and Data Communication
 - Net-centric computing
- (6) 대화기술(communication skill)
 - Technical Communication for Software Engineers
 - Technical Writing in English

- 각 과목에서 적절한 Oral Presentation 훈련
- (7) 팀워크 기술(team work)
 - Capstone Project
 - Intern Program
- (8) 사회적 임무
 - Computing Ethics and Professional Issues

위에서 (1)~(4)영역의 과목은 전공 필수로 운영을 하고, (5) 영역은 학생들의 관심에 따라서 분야 당 1~2과목을 수강하도록 하고, (6)~(8) 영역은 강한 권유를 통해 수강하도록 운영하는 것이 적절하다고 생각된다. 이러한 내용을 정리하여 그래프로 표현하면 그림 1과 같다. 이 그림은 이 특집호에 같이 실린 다른 전문 교과과정(소프트웨어 제품 개발, 시스템

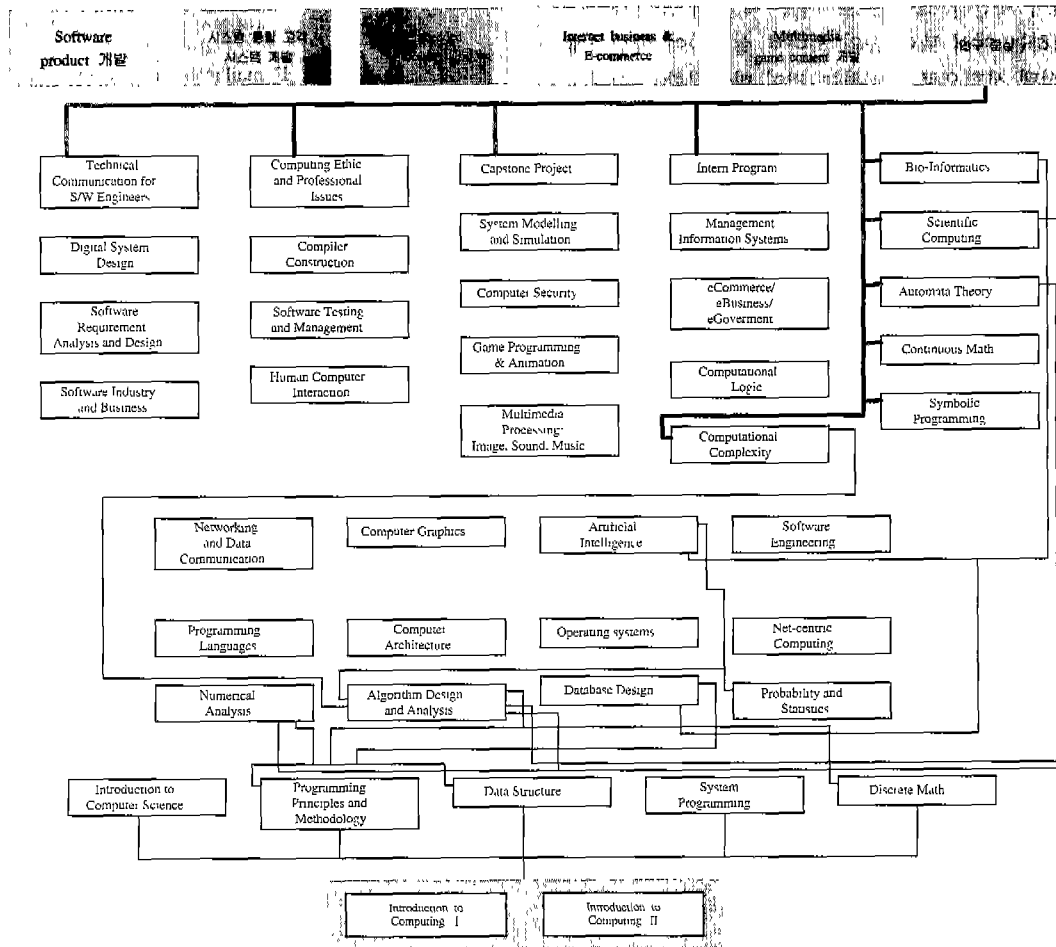


그림 1 연구-교육 인력 양성을 위한 교과과정

통합 및 고객 시스템 개발, Embedded 소프트웨어, 인터넷 비즈니스, 멀티미디어)과 같이 설계되었는데 굵은 선은 이 글이 제안한 연구-교육 교과과정의 과목을 보이고 있으며 가는 선은 선수 과목 관계를 표시하고 있다.

6. 결 론

소프트웨어 산업의 발전 속도는 놀라울 정도로 빨라서 몇 개월 주기로 새로운 응용 분야가 탄생하곤 한다. 연구-교육 분야도 이러한 상황을 충분히 고려하여 교과과정을 작성하고 운영 프로그램을 구축해야 할 것이다. 연구-교육 분야의 독특한 점은 다른 분야에 비해 소프트웨어 개발 기술보다는 이러한 기술의 토대가 되는 이론과 원리에 더 많은 비중을 두고, 수학과 과학에 대한 이해를 필요로 한다는 점이다. 이 글에서 제안한 교육 프로그램은 학과의 교과과정을 구성하는 여러 트랙 중의 하나로 설치하여 학생들에게 선택 사항으로 제공하는 것이 바람직할 것이다.

참고문헌

- [1] Avron Barr and Shirley Tessler, Software Entrepreneurism in Korea, Stanford University, December 1999.
- [2] 교수신문, "2001년 하반기 교수 임용," 2001년 10월 211호 기사.
- [3] 한겨레 신문, "위기의 대학원-서울대 박사과정 무더기 미달," 2000년 11월 23일 기사.
- [4] B. Meyer, "Software engineering in the academy," *IEEE Computer*, pp.28-35, May 2001.

[5] IEEE-ACM Joint Task Force, *Computing Curricula 2001: Volume II, Computer Science (Steelman draft)*, August 2001.

[6] 서울대학교 컴퓨터공학부 교과과정 : <http://cse.snu.ac.kr>.

[7] CMU CS Curriculum : <http://www.csd.cs.cmu.edu/bscs/currintro.html>.

오 일 석



1984 서울대학교 컴퓨터공학과 학사
 1984~1992 한국과학기술원 전산학과 석사, 박사
 1992~현재 전북대학교 컴퓨터공학과 교수
 관심분야 : 컴퓨터 비전, 문서 및 문자인식 등임.
 E-mail: isoh@moak.chonbuk.ac.kr

박 혁 로



1987.2 서울대학교 공과대학 컴퓨터공학과
 1989.2 한국과학기술원 전산학과 (전산학 석사)
 1997.2 한국과학기술원 전산학과 (전산학 박사)
 1994~1998 연구개발정보센터(현: 한국과학기술정보연구원) 선임연구원
 1999.2~현재 전남대학교 자연대학 컴퓨터정보학부 조교수

관심분야: 정보검색, 자연언어처리, 기계학습, 인공지능
 E-mail: hyukro@chonnam.ac.kr