

CS2013 지식영역의 계량화를 통한 컴퓨터과학 영역별 우선순위 설정

유병건[†] · 김자미^{††} · 이원규^{†††}

요 약

고등교육의 질 관리를 중요하게 생각한 국가들은 다양한 제도나 방법을 통해 교육의 질 강화를 진행하고 있다. 컴퓨터과학 분야에서도 고등교육의 질 관리를 위해 교육과정 표준을 구성하는 등 끊임없이 노력해 왔다. 컴퓨터과학 분야 교육과정 표준에서 언급한 지식영역의 우선순위를 파악해 보면, 내용체계 구성에 시사점을 줄 수 있을 것이다. 따라서 CS2013을 토대로 내용요소에 대한 Tier1, Tier2, Elective의 시수와 세부영역의 수를 중심으로 순위를 도출하였다. 분석결과, 가장 높은 우선순위를 지닌 지식영역으로는 Software Development Fundamentals이었다. 해당 지식영역은 CS2013에서 기초 요소이기 때문에 초급코스로 권장한다고 기술되기도 하였다. 도출되어진 영역별 우선순위가 향후 초중등 정보교육과정이나 고등 교양 정보교육과정, 교원양성기관의 정보교육과정 설정에 시사점을 줄 수 있을 것이다.

주제어 : CS2013, 지식영역, 컴퓨터과학 교육과정, 대학교 교육과정

Setting Priorities by Computer Science Area Through Quantification of CS2013 Knowledge Area

ByeongGeon Yu[†] · JaMee Kim^{††} · WonGyu Lee^{†††}

ABSTRACT

Countries that considered importance of the quality management of higher education are strengthening through various systems and methods. Field of computer science also, they have constantly made efforts, such as constituting standards of curriculum for the management of the quality of higher education. If we know the priorities of the knowledge area mentioned in the standard of curriculum of computer science, we can give implications to the composition of the content system. Therefore, we derive the ranking based on base on CS2013 the lesson time number of Tier 1, Tier 2, Elective, and detail areas for content elements. As a result, the knowledge area with the highest priority was Software Development Fundamentals. The knowledge area is recommended as a beginner course in CS2013 because it is a basic element. The priorities of these areas may provide implications for future elementary and secondary informatics curriculum, higher liberal arts informatics curriculum, Informatics curriculum of teacher training agency.

Keywords : CS2013, Knowledge Area, Informatics curriculum, University Curriculum

† 종신회원: 고려대학교 컴퓨터교육학과 박사수료
 †† 종신회원: 고려대학교 교육대학원 컴퓨터교육전공 조교수
 ††† 종신회원: 고려대학교 정보대학 컴퓨터학과 교수(교신저자)
 논문접수: 2017년 4월 17일, 심사완료: 2017년 5월 23일, 게재확정: 2017년 5월 28일
 * 본 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2016R1A2B4014471)

1. 서론

고등교육의 질 관리를 강화하기 위해 2000년대부터 현재에 이르기까지 다양한 연구가 진행되고 있다[1][2]. 고등교육의 질 관리는 국가의 발전과 국제적 경쟁력을 좌우하는 핵심 요인이라 할 수 있다[3]. 즉, 경제성장과도 관련이 있을 것으로 판단되기 때문이다[4].

질 관리를 위해 유럽 연합은 국경을 넘는 고등교육의 질에 관한 가이드라인을 제정하였다[5]. 영국의 고등교육평가원(QAA)은 대학평가인증사업을 통해 평가 결과에 따라 재정을 차등 지원하여 질 관리를 강화하고 있으며[6], 호주의 고등교육 질 관리기구(TEQSA)도 등록 표준, 범주 표준, 코스 인증 표준 등 고등교육 표준틀에 따른 평가를 통해 교육의 질을 관리하고 있다[4]. 고등교육의 질 관리 제도나 방법은 특정 학문 분야에 한정된 것이 아니며, 모든 학문 분야에서 강화되고 있는 실정이다[6].

정보통신기술(ICT)의 융합으로 이루어지는 차세대 산업혁명에서 컴퓨터과학 특히 소프트웨어(SW)에 대한 중요성이 강조되면서 SW 교육 뿐 아니라 컴퓨터과학 전반에 대한 질 관리가 이루어지고 있다[8]. 컴퓨터과학 분야의 질 관리를 위해 일본은 참조기준을 설정하였다[7][2]. Association for Computing Machinery(이하 'ACM')에서도 컴퓨터과학(Computer Science) 분야의 질을 높이기 위해 노력하고 있다. 즉, 미국이나 일본의 경우, 교육과정의 표준을 수립하여 컴퓨터과학교육의 발전에 앞장서고 있다[10][11].

한국도 2015년부터 고등교육에서 SW교육 및 교육의 질 향상을 위해 2017년 현재 20개의 SW 중심대학을 선정하여 운영하고 있다[9]. SW 중심대학의 목적은 산업 현장의 요구를 반영하고, 융합인재를 양성하기 위해 시대에 적합한 컴퓨팅 능력을 키우기 위한 것이다[9]. SW 관련 전문 인력 양성과 더불어 컴퓨터과학을 통해 문제해결력을 높이기 위한 일반교육 두 가지를 진행하고 있다.

고등교육의 컴퓨터과학 분야에 대한 미국과 일본의 노력과 달리 한국에서는 컴퓨터과학 분야만을 위한 질 관리의 노력은 미비한 편이었다. 즉,

교육에 대한 표준의 구성은 특정 학문 분야의 방향성을 제시할 수 있을 뿐 아니라[12], 교육의 계속성과 계열성 관점을 고려할 수 있을 것이다. 학문의 구조화에 따라 고등교육의 질 향상을 기대할 수 있으며, 초중등 교육 분야와의 일관성을 유지할 수 있을 것이기 때문이다[13][14].

이에 본 연구는 고등교육 분야에서 제시된 교육과정 표준이라 할 수 있는 CS2013에 나타난 지식 영역들에 대해 계량화하기 위한 목적이 있다. 일본의 J 시리즈도 표준이기는 하지만, CS2013이 최신의 흐름을 더 잘 반영하고 있기 때문에 CS2013을 선택하였다. 컴퓨터과학 분야 지식 영역에 대한 계량화는 해당 지식의 우선순위 혹은 중요도, 기초과정에서 진행해야 할 지식에 대한 방향성을 설정하는데 도움을 줄 수 있을 것이다.

컴퓨터과학에 대하여 고등교육에서는 컴퓨터과학, 초중등에서는 정보, 교육과정관점에서는 정보교육과정이라고 언급하지만, 본 연구는 고등교육 컴퓨터과학 분야의 지식영역을 주제로 하므로 '컴퓨터과학'이라고 통일하여 제시하였다.

2. 관련연구

2.1 교육과정 표준의 의미

컴퓨터과학 교육과정의 표준은 컴퓨터 분야의 학술과 교육을 목적으로 설립된 ACM에 의해 1968년에 최초로 발표되었다[15]. ACM은 컴퓨터 분야에 대한 고등교육의 질 향상에 기여하기 위해 지속적으로 연구를 진행하고 있다[11][15][16][17][18][19]. 교육과정의 표준을 구성할 때, 중요하게 고려되는 것은 내용들의 적절한 균형과 교육과정 전반에 걸친 목표 확인이다[11]. 미국은 교육과정의 표준 구성을 위해, 미국 내 대학마다 활용되고 있는 교육과정을 분석하여, 지속적인 피드백 과정을 통해 다듬어진 컴퓨터과학 교육과정의 표준을 제시하였다[11][15][16][17][18][19].

교육과정의 표준에 대한 관심은 일본의 J 시리즈에도 나타난다. 1999년 일본정보처리학회에서 '대학의 이공계 학부 정보계열학과를 위한 컴퓨터과학 교육과정 J97'이라는 이름으로 교육과정의 표준을 처음 소개하였다[23]. 2007년에는

Computing Curricula 2001 Computer Science(이하 'CC2001')의 지식체계 구성을 참고하여, J07-CS를 공표하였다[10]. J07-CS에서는 컴퓨터 과학 전공자뿐만 아니라 부전공과 교양과정에 대한 교육과정의 표준까지 제시하였다[10]. 일본에서는 모든 대학생이 배워야 하는 최소의 지식으로 구분하여, 교양 교육과정과 부전공 교육과정으로 설정한 것이다[10]. 이는 교육과정을 구성함에 있어서 모든 지식이나 태도를 전반적으로 습득할 수가 없기 때문에, 공통적으로 교육이 이루어져야 하는 최소한이 간직되어야 한다는 최소 필요량의 원리를 나타낸 것이라고 할 수 있다[20][21].

교육과정의 표준은 해당분야에서 가르치고 배워야 할 최소의 기준을 근거로, 해당분야의 지식을 체계적으로 다룰 수 있도록 구성하는 것이다[15]. 즉, 교육과정의 표준은 해당분야의 교육이 이루어질 수 있도록 지식들 간의 체계를 표현한 것이기 때문에 중요하다고 할 수 있다.

2.2 Computer Science Curricula

미국에서는 1947년 컴퓨터분야 학술 단체를 설립하여 지속적으로 컴퓨터과학교육의 발전을 위해 교육과정의 표준을 제시하였다[15]. 1968년에 발표된 교육과정의 표준에서는 컴퓨터과학의 세부 과목을 언급하였고, 1978년에 core와 elective를 구분하여 보완된 교육과정을 제시하였다[15][16]. 1991년에는 10개의 주영역과 1개의 옵션영역으로 구성하여, 과목의 형태보다 지식체계의 형태로 발표하였다[17]. 2001년에는 14개의 지식영역과 각 지식영역에 대한 세부영역을 설정하였다[18].

2005년에는 ACM과 IEEE-CS에서 CC2001(Computer Science), IS2002(Information Systems), SE2004(Software Engineering), CE2004(Computer Engineering), IT2006(Information Technology)을 한꺼번에 정리하여 Computing Curricula 2005(이하 'CC2005')를 발행하였다[22]. 2005년에 발표된 내용은 컴퓨터과학에 대한 새로운 교육과정의 표준 제시보다는 각 지식영역의 범위를 정의하였다.

컴퓨터과학 교육과정의 표준은 변화하는 사회 흐름에 따라 CC2001에서 언급한 지식영역과 세부 영역 외에 새로운 영역을 추가하거나 기존의 영

역을 수정하여 'Computer Science Curriculum 2008'을 공표하였다[19]. 2013년에는 미국 내 대학교들의 컴퓨터과학·공학 교육과정과 이수해야 할 시간을 토대로 Computer Science Curricula 2013(이하 'CS2013')을 발표하였다[11]. CS2013은 18개의 지식영역과 세부영역으로 구성되었다.

컴퓨터과학의 지식영역에 대하여 필수 이수시간과 세부영역 수를 나타내면 <표 1>과 같다[11].

<표 1> CS2013의 각 지식영역별 필수 이수시간과 세부영역 수

	지식영역	시수		세부영역 수			
		T1	T2	T1	T2	E	전체
1	AL) Algorithms and Complexity	19	9	4	4	3	7
2	AR) Architecture and Organization	0	16	0	5	3	8
3	CN) Computational Science	1	0	1	0	5	6
4	DS) Discrete Structures	37	4	6	3	0	6
5	GV) Graphics and Visualization	2	1	1	1	6	6
6	HCI) Human-Computer Interaction	4	4	1	1	8	10
7	IAS) Information Assurance and Security	3	6	3	5	7	11
8	IM) Information Management	1	9	1	3	10	12
9	IS) Intelligent Systems	0	10	0	4	9	12
10	NC) Networking and Communication	3	7	2	5	1	8
11	OS) Operating Systems	4	11	2	4	6	12
12	PBD) Platform-based Development	0	0	0	0	5	5
13	PD) Parallel and Distributed Computing	5	10	4	4	7	9
14	PL) Programming Languages	8	20	3	6	11	17
15	SDF) Software Development Fundamentals	43	0	4	0	0	4
16	SE) Software Engineering	6	22	3	9	9	10
17	SF) Systems Fundamentals	18	9	5	4	1	10
18	SP) Social Issues and Professional Practice	11	5	7	3	7	10
Total		165	143	47	61	98	163

CS2013은 Core에 대해 다음과 같은 점을 고려하여 Tier1과 Tier2로 구분하였다[11].

첫째, 각 학교 재량에 따라 컴퓨터과학교육의 필수 이수시간을 유동적으로 부여할 수 있도록 하기 위해 구분하였다. Tier1에 할당된 시간은 100% 이수를, Tier2에 할당된 시간은 100%, 90%, 80% 로 학교마다 유동적으로 이수할 수 있도록 하였다.

둘째, 컴퓨터계열 외에 타 전공에서도 컴퓨팅 개념과 기술을 교육하고 있음을 고려하여 핵심 지식 체계를 구성함에 있어 기본 개념과 응용 개념에 따라 가이드를 제공하고자 Tier1과 Tier2로 구분하였다.

필수로 이수해야하는 영역들 외에 심화 학습을 위한 선택 이수 영역도 존재한다. <표 1>에서 PBD영역을 살펴보면, Tier1과 Tier2에 대한 시수가 0임을 알 수 있다. 이는 PBD영역에 포함되는 세부영역들 모두 필수 이수가 아닌, 심화 학습을 위한 Elective만 존재한다는 뜻이다. GV의 경우, Tier1은 2시간, Tier2는 1시간이 할당되어 있지만, 전체영역 수와 Elective 주제 수가 동일하다는 것은 GV의 모든 세부영역이 Elective를 포함한다는 뜻이다. GV영역과 PBD영역의 세부영역 주제를 살펴보면 <표 2>와 같다.

<표 2> GV와 PBD의 세부영역

세부영역		T1	T2	E
G V	Fundamental Concepts	2	1	Y
	Basic Rendering			Y
	Geometric Modeling			Y
	Advanced Rendering			Y
	Computer Animation			Y
	Visualization			Y
P B D	Introduction			Y
	Web Platforms			Y
	Mobile Platforms			Y
	Industrial Platforms			Y
	Game Platforms			Y

CS2013에는 추가된 영역도 존재한다. SF 영역의 경우, 근본적으로 다른 지식영역들과 상호 연결되어 있어 각 지식영역들에 대한 개념을 보완해 주어 이해할 수 있도록 입문과정과 시스템과정으로 구성되었다. 입문과정은 컴퓨터과학의 개

념 이해와 소프트웨어 개발의 기본 사항을 주제로 정하여 다룬다. 시스템 과정은 시스템의 기초를 습득하는 것으로써, 컴퓨터 아키텍처, 운영체제, 네트워크, 분산 시스템을 연계하여 학습할 수 있는 기반이 된다. 아키텍처를 살펴보면, 2001년과 2008년에 발표한 교육과정의 표준에서 36시간이라는 많은 이수시간을 차지하였다. 그러나 CS2013에서는 아키텍처의 기본 개념과 고급 과정을 분류하여, 각각 SF영역과 AR영역에 할당하였기 때문에 앞서 발표한 교육과정의 표준과 다른 시간 할당을 보여준다.

이와 같이, 컴퓨터과학 교육과정의 표준은 변화하는 사회의 흐름에 맞춰 개정을 진행해 왔다 [11][15][16][17][18][19]. 개정하는 과정 중에 컴퓨터과학에 대한 지식체계를 만들었고, CS2013에서는 지식영역을 제각각 구분하여 학습하기보다는 지식영역끼리 연계학습이 가능하도록 표현까지 하였다.

3. 우선순위 설정 방법

3.1 우선순위 설정의 고려사항

CS2013은 컴퓨터과학 분야의 지식영역을 설정하였고, 미국 내 각 대학에서 실시하고 있는 해당 지식에 대한 수업 시수와 수준 등을 제시하고 있다. 본 연구는 CS2013에 근거하여 다음과 같은 방법으로 우선순위를 설정하였다.

첫째, 이수시간의 고려이다. 수업에서의 시간은 ‘얼마나 많은 학습 기회가 주어졌는가’에 대한 것으로 해당 지식이 전체 지식 대비 중요하기 때문인 것으로 볼 수 있다[24]. CS2013에 대비해 보면, 지식 영역 중 Tier1에 많은 시간이 배정된 경우, 모든 컴퓨터과학 교육과정의 필수로 제시되어 있으며, 입문 과정에서 해당 내용을 배울 가치가 높다고 할 수 있다. 반면, Elective의 경우 이수시간이 제시되지 않았으며, 배우는 내용 또한 심화된 것이기 때문에 시수를 기준으로 하는 검증에서는 배제하였다.

둘째, 세부영역 수에 대한 고려이다. 지식 영역의 세부영역은 Tier1, Tier2에 시간이 배정되어 있고, Elective는 유무만을 제시하고 있다. 본 연

구는 지식 영역의 세부영역의 수가 많다는 것의 의미를 파악하였다.

세부 영역의 경우 <표 2>의 GV영역과 같이 하나의 세부 영역에 Tier1, Tier2, Elective가 중복적으로 포함될 수 있다. 즉, 'Fundamental Concepts' 은 Tier1에서 2시간, Tier2에서 1시간 그리고 시간적 여유가 있고, 심화의 필요가 있다면, Elective에서 각각 다른 내용을 학습하도록 권장하고 있다. 지식영역에 따라 세부영역의 수를 고려할 때는 중복으로 모두 포함해야 함을 알 수 있다.

셋째, 이수시간과 세부영역을 동시에 고려할 필요가 있다. 세부영역이 많으면, 이수시간이 상대적으로 많아서 우선적으로 배워야 할 지식인지를 판단하기 위해서이다.

예를 들면, SDF영역은 세부영역 4개, 43시간이 배정되었다. PL영역은 세부영역 17개, 28시간 배정되어 있다. PL영역은 18개의 지식 영역 중 가장 많은 세부영역으로 구성되었다. PL은 매우 다양한 언어로 작업할 것에 대비하는 것이며, 소프트웨어 개발자들이 이해하고 있어야 할 다양한 프로그래밍 모델에 대한 것을 다루고 있다. 즉, 소프트웨어 개발자를 위한 중급수준의 알고리즘 등에 대한 내용인 PL은 우선되어야 하는 지식은 아니라고 할 수 있다.

넷째, CS2013에서 제시된 코스내용을 고려하였다. CS2013에서 SDF는 프로그래밍과 알고리즘의 기본 내용을 가르치므로 초기 코스, PL은 중급, SF는 분산시스템이나 병렬시스템, 컴퓨터 아키텍처 등에 대한 내용을 포함한다. SF는 입문과정과 중급을 포함하며 복잡하지는 않아도 기초개념들이 어떻게 통합되어야 하는지에 대한 측면을 다룬다.

다섯째, 앞에서 제시한 4가지를 토대로 객관적인 산출을 위해 19개의 수식을 만들어 8개의 범주로 구분하고 타당한지 검토하였다.

3.2 우선순위 설정을 위한 수식 도출 및 절차

본 연구는 앞에서 제시한 다섯 단계에 따라 CS2013 지식영역에 대한 우선순위를 도출하였다. 도출과 더불어 우선순위에 따르는 비중을 산출하

기 위해 수식을 활용하였다.

19개의 수식을 계산하여 8개의 범주 중에 가장 타당한 결과를 보여준 경우는 Elective를 배제하고, Tier2 시수의 80%와 Tier1 시수의 100%를 활용하여 도출한 우선순위였다. 해당 우선순위 도출 수식을 표현하면 다음과 같다.

$$\frac{KA_i(T1_t) + (KA_i(T2_t) \times 0.8)}{T1_t + (T2_t \times 0.8)} \times 100 = KA_i(W)$$

(단, $1 \leq i \leq 18$)

위의 수식은 <표 1>에서 제시한 CS2013 지식영역의 순서를 1부터 18까지로 보고, i 값의 범위를 ' $1 \leq i \leq 18$ '로 지정하였다. 즉, 18개의 지식영역을 ' $KA_1 \sim KA_{18}$ '로 작성하였다. Tier1의 시수는 ' $T1_t$ ', Tier2의 시수는 ' $T2_t$ ', 계산한 결과 값으로 도출된 각 지식영역별 비중은 ' W '로 하여 수식을 표현하였다.

본 연구는 CS 관련 지식의 우선순위를 설정하는 데 있어서 다음과 같은 절차로 진행하였다.

첫째, 우선순위 설정을 위해 다양한 관점을 고려하였다.

둘째, 우선순위 설정 과정에서 수치화를 통해 중요도의 정도를 제시하였다. 수치화는 다양한 수식을 산출하고, 해당 수식을 통해 제시된 내용을 우선순위 설정 방법을 통해서 우선순위와 비중을 산출하였다.

셋째, 지식영역의 우선순위와 비중에 대해서는 컴퓨터과학 전문가 5인의 검증을 통해 내용 타당도를 확보하였다.

4. 컴퓨터과학 내 지식의 우선순위

4.1 CS 지식영역의 우선순위

컴퓨터과학 분야에서 제시된 지식영역에 대한 우선순위는 앞에서도 언급한 바와 같이 다양한 관점과 수식을 통해 확인하고 검증하였다. 최종적으로 전문가들에 의해 타당성 검증 설정을 토대로 컴퓨터과학 내 영역별 우선순위를 도출하였다. 우선순위 설정의 결과는 <표 3>과 같다.

<표 3> 컴퓨터과학 지식영역의 우선순위와 비중

우선순위	지식영역	비중
1	SDF	15.39
2	DS	14.39
3	AL	9.38
4	SF	9.02
5	PL	8.59
6	SE	8.45
7	SP	5.37
8	PD	4.65
9	OS	4.58
10	AR	4.58
11	NC	3.08
12	IM	2.93
13	IS	2.86
14	IAS	2.79
15	HCI	2.58
16	GV	1.00
17	CN	0.36
18	PBD	0.00

CS2013 지식영역의 우선순위 결과, 1순위는 SDF이며, 2순위는 DS, 3순위 AL의 순서였다. 반면, GV, CN은 우선순위가 낮았으며, PBD는 지식영역의 우선순위에서는 매우 미약함을 확인하였다.

PBD(Platform-based development)는 일반적인 지식의 범위를 커버하는 SDF와 같은 영역과 다른 특성이 있다. 특별한 플랫폼에서 강조되고 필요로 하는 특정 지식에 근거하기 때문에 Tier1이나 Tier2에도 포함되지 않고, Electives에만 모든 세부 영역이 포함되어 있음을 알 수 있다.

CN(Computational Science)은 CS 분야와 직접적인 관련보다 CS적용 분야라고 할 수 있다. 즉, SDF의 Algorithms and Design, Development Method 그리고 AR의 Machine representation of Data, Memory system Organization and Architecture 등 SDF, AR 등의 영역과 관련이 있지만, 기본 개념이 아닌 응용 분야에서 필요로 하는 지식 및 활용을 습득하기 위한 지식이다.

4.2 CS 지식영역의 우선순위의 의미

본 연구에서 도출한 지식의 우선순위와 비중은

비중을 나타내는 각 숫자가 해당 지식의 기초적인 측면에서 무엇보다 우선되어야 함을 의미하고 있다. 즉, 각 지식영역의 순서와 비중은 고등교육의 시작에서 컴퓨터과학 분야에서 어떤 지식으로 접근해야 하는지를 제시하고 있다고 할 수 있다. CS 지식영역에 대한 우선순위와 의미를 구체적으로 분석하면 다음과 같다.

첫째, 용어의 유사성이 전체 의미나 깊이의 유사성은 아니라는 점이다. CS 분야의 경우, 동일한 용어를 사용한다고 해도 지식의 깊이는 크게 다른 경우가 많다. 앞에서도 언급한 바와 같이 PL(Programming Languages)의 경우, 초중등 교육에서는 교육용 프로그래밍 언어(Educational Programming Languages: EPL)라는 용어를 사용한다. 프로그래밍 언어라고는 하지만, 해당 쓰임과 용어의 깊이는 전혀 다르다. 프로그래밍 언어에 대해 더 구체적이고 전문적인 용어로서의 쓰임이라는 것을 CS2013의 여러 곳에서 제시하고 있는 것도 이와 같은 이유에서 이다.

둘째, CS2013의 지식영역 구성 및 지식에 대한 설명이 어떤 교육 내용을 통해 준비되어야 하는가에 집중하고 있음을 파악해야 할 것이다. 2015 개정 교육과정에서 정보과는 컴퓨팅 사고력을 정보문화 소양, 협력적 문제해결력과 함께 핵심 역량으로 제시하고 있다. 컴퓨팅 사고력과 유사한 용어로 보이는 CN영역은 컴퓨팅 사고력이 발현되어야 하는 집합체이다. 기존의 컴퓨팅 지식을 토대로 다른 분야와 융합하고 문제를 해결해 갈 수 있어야 할 것을 영역 전체의 목표로 삼고 있기 때문이다.

초중등 교육에서 현재의 17시간, 34시간 그리고 선택 68시간을 통해 컴퓨팅 사고력을 육성할 수 있을 것인지에 대한 의문이 드는 부분이다. CN영역은 Tier1에 1시간과 Elective로 구성되어 있다. 초중등 교육을 통해 고등교육에서 요구하는 능력을 습득할 수는 없다. 그러나 유사한 용어의 늪에 빠지지 않기 위해서는 해당 영역의 지식을 통해 무엇을 습득할 수 있을 것인지, 그리고 무엇을 습득해야 하는지를 고려할 때, 컴퓨팅 사고력의 향상을 위해 현재의 교육 시간은 타당한 것인지 등에 대한 광범위한 고민이 필요할 것으로 판단된다.

셋째, 초중등 교육으로의 확산을 위해서 지식의 구조 측면은 벤치마킹이 가능하지만, 보다 구체화된 노력이 요구된다. CS2013은 국제적으로 201명의 위원회 위원이 참여하였고, 그 중 10,000명 이상의 학생을 둔 대학이 44.5% 참여하였다. 즉, 100여개가 넘는 대학의 교육과정에 근거한 교육과정 표준인 만큼 고등교육과정으로서는 의미가 있다. 그러나 우선순위가 높다고 해서 기초적인 지식으로 바로 사용할 수 있음을 의미하는 것은 아니라는 점이다. 예를 들면, 초중등 학교에서 CS를 가르치기 위한 지식영역을 논의하기 위해서는 보다 정제된 데이터가 요구된다는 점이다.

정제된 데이터는 고등 교육과정을 초중등 수준으로 어떻게 내릴 것인지, 지식의 구조 속에서 핵심 지식을 CS2013에 둘 경우, 어느 정도 추상화하여 학생들에게 가르칠 수 있을 것인지, 어떻게 가르칠 것인지, 그리고 해당 내용을 이해하기 위해 학생들에게 요구되는 시간은 어느 정도인지 등에 대한 것이다. 용어의 유사성만을 고려할 것이 아니라 지식의 구조를 이해하고, 해당 지식영역에 대한 깊은 이해를 바탕으로 초중등 교육과정에 도입할 필요가 있을 것이다. 예를 들면, 지식의 수준에 따르는 교육 목표를 설정하고, 어떤 수준까지 지식을 제공할 것인지에 대한 내용을 설정하고, 설정된 지식을 통해 학습자들이 어떤 경험을 할 수 있는지를 고려한 다음 최종적으로 평가에 대한 측면까지를 의미한다.

교육을 위해 요구되는 지식은 지식 자체의 가치도 중요하지만, 지식을 어떻게 내재화하고, 앞으로 습득하게 될 지식과 연계할 것인지에 대한 충분한 고려를 포함해야 한다. 이는 교육과정 구성에서 지식을 선택할 때의 관점이다. 지식의 깊이와 위계를 충분히 고려하여 학습자들의 수준에 적합한 교육이 진행될 수 있도록 해야 할 것이다.

넷째, 시수에 대한 시사점을 고려할 수 있다. 프로그래밍 언어 영역의 경우, 최소 15개 대학에서 프로그래밍 언어라는 이름으로 강의가 개설되어 있으며, 컴파일러의 명칭으로 유사한 강의가 5개 대학에 개설되어 있다. 각 대학들은 최소 시간이 20시간 정도이며, 최고 38시간 이상을 배정하였다. 35시간을 개설한 브라운 대학(Brown University)의 경우, 프로그래밍 언어 소개의 과정

으로 개설하고 35시간을 배정했다. 과목 소개에는 학습자가 본 강좌를 이해하기 위해 일주일에 최소 10시간에서 20시간 이상을 투자해야 한다고 제시하였다.

교육 목적을 달성하기 위해 요구되는 시간은 수업 시간과 별개임을 의미한다. 따라서 CS 지식영역에 대한 고려, 교육과정 구성에의 도입 등을 고려할 때는 수업에서 제시된 시수 뿐 아니라 해당 내용을 이해하는데 학습자에게 요구되는 시간을 충분히 고려해야 할 것이다.

5. 결론

미국의 컴퓨터과학 교육과정의 표준이 2013년까지 이른 것은 지속적인 피드백과정을 통해 대학들의 교육과정을 분석하여 지식영역의 체계를 구성하였기 때문이다[11]. 즉, 단발성으로 구성하지 않고, 고등교육의 질 향상을 위해서 끊임없이 노력을 한 것이다.

2017년 현재 한국에서도 컴퓨터과학에 대한 고등교육의 질 강화를 위해 SW중심대학 사업을 진행하고 있다. 교육에서 교육과정 표준을 구성하는 것은 어려운 일이다. 기존의 교육과정 표준이 존재한다 하더라도 학습자의 선행 지식에 대한 충분한 이해가 수반되어야 되기 때문이다.

학습자가 배제된 교육과정 표준이 어떤 의미를 갖게 되는가를 고민할 필요가 있다. 고등교육의 수준에서만 질을 관리하고 수준 높은 교육을 진행하는 것으로는 충분한 질 관리가 이루어질 수 없음을 간과해서는 안 될 것이다. CS2013에서 제시하는 지식영역과 수준은 학습자에게 매우 많은 시간과 노력을 요구하고 있다. 기초가 부족한 학습자라면, 해당 학문에 대한 접근성이 낮지 않음을 알 수 있다.

교육에서, 그리고 교육과정에서 질을 관리하기 위해서는 교육과정 본연의 계열성과 계속성의 관점이 유지되었을 때이다. 즉, 고등교육에서 교육과정 표준이 존재하기 위해서는 초중등학교에서 교육과정 표준이 적용될 수 있어야 한다. 고등교육 수준에서만 높은 질을 유지하는 것은 불가능하기 때문에 사회 저변의 광범위한 인력이 해당 지식에 대한 이해를 갖출 수 있도록 하는 노력이

요구된다.

본 연구는 CS2013에서 제시하는 지식영역에 대한 이해를 바탕으로 각 지식영역에 대한 우선순위를 설정하였다. 우선순위를 비중으로 제시하여, 해당 지식영역의 우위를 설명하였다. 그리고 한국에서 정보교육을 통해 육성하고자 하는 핵심역량 중 하나인 컴퓨팅 사고력이 2015 개정 교육과정에서 제시하는 시수만으로 육성될 수 있을 것인지에 대한 의문을 제기하였다.

2015 개정 교육과정은 2017년 현재 초등학교 1, 2학년이 적용되고 있을 뿐이기 때문에 ‘정보과’는 현장 속으로 들어가지 않은 상황이다. 아직 시작도 하지 않았지만, 부족함을 느끼게 하는 교육과정이 어느 정도 실효를 거둘 수 있을지에 대한 의문을 제기하지 않을 수 없다.

이상과 같이 본 연구는 CS2013에 대한 구체적인 이해를 바탕으로 교육과정 표준의 의미를 살펴보고, 향후 고등교육에서 학문적 질 관리를 위해서는 초중등 교육부터 개선이 이루어져야 함을 제시하였다. 본 연구는 많은 교과에서 초중등 교육과 고등교육의 연계를 강조하는 가운데 사회적 필요에도 불구하고, 교과의 역할에 맞는 시수를 확보하지 못하고 있는 초중등 정보에 대한 시사점을 제공했다는 데 의의가 있다.

참 고 문 헌

- [1] 김영화, 백정하, 이윤미, 서정화, 신현석 (2009). **고등교육의 이해 이론과 실제**. 서울: 학지사.
- [2] 이원규(2016). 대학교육의 분야별 질 보증을 위한 교육과정 편성상의 참조기준 정보학 분야. **컴퓨터교육학회 기고문**. 19(3).
- [3] 신동은, 최금진(2011). 한국, 일본, 대만의 공학교육인증 평가체제 및 평가기준 비교. **교육행정학연구**. 19(1), 319-345.
- [4] 김정희, 주동범, 박현주(2012). 호주 고등교육 질관리기구(TEQSA)의 주요 특징과 시사점. **비교교육연구**. 22(5), 127-154.
- [5] 채재은, 이병식(2007). 고등교육 질 보장 (Quality Assurance) 접근 방법에 대한 비교 연구: 미국, 호주, 핀란드 사례를 중심으로. **교육과학연구**. 38(1), 215-235.
- [6] 최금진(2015). 영국의 고등교육과 질 관리 평가에 관한 연구. **비교교육연구**. 25(4), 63-91.
- [7] 萩谷昌己(2014). 정보학을 정의한다 - 정보학 분야의 참조기준. **일본정보처리학회**. 55(7), 734-743.
- [8] 한경 경제용어사전. <http://dic.hankyung.com>
- [9] 미래창조과학부. <http://www.msip.go.kr>
- [10] 일본정보처리학회(2009). **표준교육과정 컴퓨터과학 J07-CS**. 일본정보처리학회 컴퓨터과학교육 위원회.
- [11] ACM IEEE-CS(2013). *Computer Science Curricula 2013*. US: ACM IEEE Computer Society.
- [12] 김자미, 이원규(2016). CSTA 2003과 2011 비교를 통한 한국의 정보교육과정 표준에 대한 시사점. **컴퓨터교육학회**. 19(1), 41-51.
- [13] 서울대학교 교육연구소(2011). **교육학 용어사전**. 서울: 하우동설.
- [14] 김자미, 이원규(2014). 브루너의 이론에 근거한 인도의 정보교육과정 고찰. **컴퓨터교육학회**. 17(6), 59-69.
- [15] ACM(1968). *Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science*. Communications of the ACM CACM Homepage archive. 11(3), 151-197.
- [16] ACM(1979). *Curriculum 78: Recommendations for the undergraduate program in computer science - a report of the ACM curriculum committee on computer science*. Communications of the ACM CACM Homepage archive. 22(3), 147-166.
- [17] ACM(1991). *Computing Curricula 1991*. Communications of the ACM. 34(6), 68-84.
- [18] ACM & IEEE-CS(2001). *Computing Curricula 2001 Computer Science*. US: The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery.

- [19] ACM IEEE-CS(2008). *Computer Science Curriculum 2008*. US: ACM IEEE Computer Society.
- [20] 구병두 외(2015). **교육과정**. 서울: 양서원.
- [21] 김상원(2015). **소요 교육학논술**. 서울: 은하인.
- [22] ACM IEEE-CS(2005). *Computing Curricula 2005*. US: ACM IEEE Computer Society.
- [23] 일본정보처리학회(1999). **대학의 이공계학부 정보계열학과를 위한 컴퓨터과학교육 교육과정 J97**. 일본정보처리학회 컴퓨터과학교육 위원회.
- [24] Hamischfeger, A.(1985). Active learning time, In T. Husen & T. N. Postlethwaite(eds.). *The International Encyclopedia of Education. 1.* 42-46.



유 병 건

2010 군산대학교
컴퓨터정보공학과(공학사)
2012 고려대학교
컴퓨터교육전공(교육학석사)
2014 고려대학교 컴퓨터교육학과 박사수료
관심분야: 정보교육, 프로그래밍교육, 놀이기반
E-Mail: byeonggeon.yu@inc.korea.ac.kr



김 자 미

1992 이화여자대학교
교육학과(문학사)
1995 이화여자대학교
교육학과(문학석사)
2011 고려대학교 컴퓨터교육학과(이학박사)
2011~2015 고려대학교 컴퓨터학과 연구교수
2015~현재 고려대학교 교육대학원
컴퓨터교육전공 조교수
관심분야: 정보교육, 교육과정평가, 이러닝
E-Mail: celine@korea.ac.kr



이 원 규

1985 고려대학교
영어영문학과(문학사)
1989 츠쿠바대학 이공학연구과
(공학석사)
1993 츠쿠바대학 공학연구과 전자·정보공학
전공(공학박사)
1993~1995 한국문화예술진흥원 문화정보본부
책임연구원
1996~2014 고려대학교 사범대학 컴퓨터교육과 교수
2014~현재 고려대학교 정보대학 컴퓨터학과 교수
관심분야: 정보교육, 정보표현, 정보관리, 교육정책
E-Mail: lee@inc.korea.ac.kr