

SW 교육의 올바른 방향에 대한 고찰

- Computational Thinking vs. Coding -

양 단 회*

◆ 목 차 ◆

- | | |
|-----------|--------------|
| 1. 서 론 | 4. CT 교육의 방향 |
| 2. CT 개념 | 5. 결 론 |
| 3. 국내외 상황 | |

1. 서 론

오바마 미국 대통령은 올해 1월 모든 초·중·고 학생들에게 컴퓨터과학을 가르치겠다는 ‘모두를 위한 컴퓨터과학(Computer Science for All)’ 추진에 대한 주간 담화를 발표하였다. 모든 아이들이 미래 직업에 대한 준비를 할 수 있도록 단순히 컴퓨터를 사용하는 것에 그치지 않고, 분석 기술과 코딩 역량을 향상시켜 혁신 경제에 힘을 불어넣어야 한다는 것이다. 그리고 새로운 경제에서 컴퓨터과학은 선택의 문제가 아니고, ‘읽기’, ‘쓰기’, ‘셈하기’(3Rs; Reading, wRiting, aRithmetics)와 함께 모두가 기본적으로 갖추어야 할 기초 소양이라고 공표하였다[1].

페이스북 창업자 마크 저커버그는 “15년 후 프로그래밍 교육을 보면서 왜 더 일찍 시작하지 않았는지 의아해할 것이다.”라고 프로그래밍 교육의 중요성을 언급했으며, 넷스케이프를 개발한 마크 안드레센은 “앞으로 SW가 세상을 삼킬 것”이라고 단언하였다[2]. 2006년 카네기 멜론 대학의 Jeannette Wing 교수는 읽기, 쓰기, 셈하기와 같은 기본 소양에 CT(Computational Thinking)를 추가해야 한다고 주장하였다[3].

컴퓨터과학 교육, 코딩 교육, 프로그래밍 교육, SW 교육, CT 교육. 이렇게 사용된 단어는 다르지만 이들이 던지는 메시지는 하나다. 모두들 창의성과 문제 해결 능력을 배양시켜야 한다는 것이다. 그러나 우리

는 이 용어들을 분명히 구분하여 사용할 필요가 있다. 그래야 요새 교육부와 미래부에서 이야기하는 SW 교육을 위해 무엇을 배우고 가르쳐야 할지를 규정할 수 있기 때문이다.

그런데 SW 교육을 코딩 교육이라고 오해하고 있는 사람들이 많다. 특히 대학현장에서 이러한 현상이 있는데 이것은 아마 컴퓨터과학과, 소프트웨어학과 등 이미 SW 교육 관련 학과가 존재하며, 이들 학과들의 정체성을 비전공자가 피상적으로 보기엔 프로그래밍을 배우는 학과로 인식되고 있기 때문일 것 같다. 이것은 영어영문학과를 영어회화나 영작문을 주로 배우는 학과로 일반인들이 인식하고 있는 것과 비슷하다. 그래서 본고에서는 SW 교육과 관련된 용어들의 개념을 명확히 하고, 올바른 SW 교육의 방향에 대해 살펴 보려고 한다.

2. CT 개념

2.1. CT 용어 번역과 정의

Computational Thinking(CT)을 어떻게 번역할 것인가? Computational Linguistics, Computational Biology, Computational Chemistry 등에서 ‘Computational’은 ‘컴퓨터상에서 구현 가능한, 컴퓨터로 시뮬레이션할 수 있는’의 의미로 ‘계산’ 혹은 ‘전산’이란 용어로 번역된다. 그런데 이에 따라 Computational Thinking을 ‘계산

* 평택대학교 컴퓨터학과 교수

(적) 사고'로 번역하면 '손익관계를 따지는 사고'로 오인될 수 있어 'Computational' 대신 '정보를 처리하는'의 의미를 갖는 'Computing'을 차용하여 '컴퓨팅적/컴퓨팅 사고'로 번역되고 있다. 그러나 Computational Linguistics가 전산 언어학으로 번역되는 사례를 따르면 Computational Thinking은 '전산적 사고'가 가장 합리적인 번역으로 보인다.

Jeannette Wing은 CT의 핵심요소를 추상화(abstractio)와 자동화(automation)로 보았다. 추상화는 문제해결을 위해 문제를 분해하거나 중요한 부분을 끌어내는 것 등을 통해 해결해야 할 문제의 복잡성을 효과적으로 줄이는 것이다. 자동화는 추상개념이 수행해야 할 일들을 컴퓨터가 수행할 수 있도록 해결과정을 알고리즘화하는 과정으로 규정하였다[3]. 그리고 2010년에 Jeannette Wing은 CT를 주어진 문제가 정보처리 Agent, 즉 컴퓨터에 의해 효과적으로 수행될 수 있는 형태로 표현될 수 있도록 문제와 그 해결책을 형식화하는 것과 관련된 사고과정이라고 재정의하였다[4]. 이 정의는 '전산적 사고'라는 번역과 잘 호응된다.



(그림 1) CT 개념[5]

미국 국제교육기술협회(ISTE: International Society for Technology in Education)와 컴퓨터과학 교사협회(CSTA: Computer Science Teachers Association)는 CT를 문제 해결에 컴퓨터나 다른 도구를 사용할 수 있도록 문제를 구성하는 것, 논리적으로 데이터를 조직하고 분석하는 것, 모델링이나 시뮬레이션 등의 추상화를

통해 데이터를 표현하는 것, 알고리즘적 사고를 통해 해결책을 자동화하는 것, 목표 달성을 위한 해결책을 식별하고 분석하고 적용하는 것, 이러한 문제해결과정을 다양한 문제들로 일반화하고 전환하는 것을 포함하는 문제해결과정이라고 정의하고 있다. 최숙영[5]은 (그림 1)에서처럼 CT를 문제해결을 위한 것으로 추상화와 자동화가 그 핵심이고, 다시 분석적 사고, 논리적 추론, 알고리즘적 사고, 창의적 사고, 인지적 스킬이 요구되는 것으로 보았다.

결론적으로 CT는 전산적 사고, 컴퓨팅적 사고, 컴퓨터과학적 사고로 번역되고 있으며, 컴퓨터과학의 이론, 기술, 도구를 활용하여 우리가 일상에서 부딪치는 복잡한 문제를 정형화하고 해결하기 위한 사고라고 정의할 수 있다.

2.2. Jeannette Wing이 CT에서 강조한 5가지 사고

Jeannette Wing이 CT의 주요 요소로 강조한 5가지 사고는 다음과 같다[3].

- ① 개념화(Conceptualizing)
단순히 코딩하는 시각으로 접근하지 않고, 분석, 간략 설계, 상세 설계, 코딩 등 여러 단계의 추상화 시각에서 접근할 수 있는 사고
- ② 분해(Decomposition)
복잡하고 어려운 문제를 잘게 분할하여 정복(divide and conquer)하는 방법으로 해결할 수 있는 사고
- ③ 추상화(Abstraction)
복잡한 문제의 공통적인 부분을 인식하여 파악할 수 있는 사고
- ④ 재귀적 사고(Recursive Thinking)
동일한 해법을 문제 해결이 될 때까지 반복 적용할 수 있는 사고
- ⑤ 병렬 처리(Parallel Processing)
문제를 병행적으로 파악하여 처리할 수 있는 사고

2.3. K-12의 CT 교육내용

미국에서는 CT를 모든 분야에 걸쳐 접근되고 적용될 수 있는 하나의 문제 해결 방법으로 간주하고, K-12의 각 단계에서 CT를 체계적으로 교육시키기 위해 (표 1)과 같이 9가지의 핵심요소와 핵심어를 정립하였다[5,6].

(표 1) K-12에서 CT 요소와 핵심어(6)

CT Components	Keywords
Data Collection	Data Source
Data Analysis	Statistical Calculations
Data Representation	Data Structures
Problem Decomposition	Objects, Methods
Abstraction	Procedure, Encapsulation
Algorithms & Procedures	Algorithm, Implementation
Automation	-
Simulation	Threading, Pipelining
Parallelization	Algorithm Animation, Parameter Sweeping

2.4. CT 교육의 필요성

- ① CT 교육은 미래 인재들이 새로운 경제에서 생존과 경쟁력을 갖추기 위한 창의력, 분석력, 논리력 등 핵심 역량을 키워준다.
- ② 단순히 컴퓨터를 사용하는 것에 그치지 않고 분석 기술과 프로그래밍 역량을 향상시켜 혁신 경제에 힘을 불어넣을 수 있게 한다[1].
- ③ 코딩이나 프로그래밍 기술과는 달리 전공 분야에 관계없이 컴퓨터과학의 원리를 활용해 상황을 분석하고 재구성해 이를 해결하기 위한 일련의 구조를 만들어 주어진 문제를 해결할 수 있는 보편적인 시각을 제공한다.

3. 국내외 상황

모바일로 촉발된 전 세계 IT 문화는 이제 보편적인 HW의 보급과 정체되어 가는 IT 물리기술이 한계에 다다르자, SW에서 그 답을 찾기 시작했고, 실제로 하드웨어 보다 SW에 몰입한 회사들의 경쟁력과 생산성이 높아지기 시작했다[2].

SW 교육은 일반 산업계에서 SW 개발자를 양성하기 위한 교육을 의미하나, 최근 SW가 미래 산업의 중요한 분야로 대두되면서 모든 학생들에게 가르쳐야 하는 교육으로 부각되고 있다. 미국과 영국을 비롯한 선진국에서도 SW 교육을 강화하고 있으며, 모든 학생들을 위한 SW 교육의 목표를 CT 역량 강화로 설정하고 있다[7,8].

주요 선진국들이 SW 교육에 적극 나서는 이유는 그만큼 경제·산업 영역에서 SW 산업 비중이 커지고 있기 때문이다. 이미 전 세계 IT 시장은 2002년을 기점으로 SW 산업 규모가 HW 산업 규모를 넘어섰으며 이 차이는 계속 더 벌어질 전망이다[2,8].

국운이 걸린 만큼 각국의 SW 교육도 바뀌고 있다. SW 교육의 핵심은 문제가 주어지면 어떻게 해결할지 절차를 구상하고, 그 절차에 따라 차근차근 해결해 나가도록 논리적 사고력을 발휘하는 CT 역량을 배양시키는 것이다.

이미 영국과 미국 등 선진국에서는 CT를 정규 교과과정으로 채택하여 새로운 교육 트렌드로 자리 잡았다. 카네기 멜론 대학교는 MS와 함께 CT센터를 설립하고 학생들에게 프로그래밍 구조화, 데이터 조직화 등으로 구성된 강의를 제공하고 있으며, 하버드대학교도 CT에 기반을 둔 문제해결과정을 가르치는 등 세계 유수의 대학에서 CT 과정을 기초과정으로 도입 중이다[5].

우리도 SW 교육 열기가 뜨겁다. 교육부와 미래부는 2015년 7월, 정부의 핵심개혁과제인 ‘창조경제 핵심성과 창출’을 앞당기고 SW 중심사회에서의 주역인 SW 인재 양성을 위해 ‘SW중심사회를 위한 인재양성 추진계획’을 보고하였다. 이를 통해 초·중·고 및 대학에 이르는 SW 교육의 기본 틀을 마련하고, 대학 SW 교육의 혁신을 추진하고 있다[9].



(그림 3) SW 교육 중점 추진전략(9)

정부는 SW 중심사회 구현을 위해 중학생은 2018년부터, 초등학생은 2019년부터 SW 교육을 필수화하도록 교육과정을 개편했다. 이제 SW 교육은 거스를 수 없는 대세가 되어 전 세계의 교육 패러다임이 급격히 바뀌고 있다.

4. CT 교육의 방향

4.1. 컴퓨터과학에서의 유사 과목

컴퓨터과학에서 CT와 가장 유사한 과목으로 고려할 수 있는 후보로는 프로그래밍, 자료구조, 알고리즘, 소프트웨어공학 과목을 대표적으로 들 수 있다. 여기서 각 과목의 내용과 특성을 살펴보겠다.

4.1.1. 프로그래밍

영어를 가르친다고 영문학 작품을 잘 창작할 수 있는 것은 아니다. 이와 마찬가지로 프로그래밍 언어를 가르친다고 소프트웨어를 잘 개발할 수 있는 것은 아니다. 그리고 어떤 언어를 가르친다는 것은 그 언어의 어휘와 문법을 습득시켜 의사소통을 할 수 있게 하는 것으로 일반적으로 인식된다.

컴퓨터과학에서도 프로그래밍 언어의 문법을 가르쳐 기본적인 구문을 작성할 수 있게 만드는 것이 프

로그래밍 수업의 목표이다. 그런데 그것으로 끝이다. 영어학 수업에서 영문학 작품을 어떻게 잘 작성할 것인가를 가르치지 못하는 것처럼, 프로그래밍 수업에서도 소프트웨어를 어떻게 잘 개발할 것인가에 대한 내용은 없다.

그러므로 CT 교육에서 특정 프로그래밍 언어를 깊숙이 숙달시키려는 시도는 바람직하지 않다. CT 교육이 전문적인 소프트웨어 개발자를 양성시키려는 교육이 아닌 만큼 프로그래밍 교육용 도구를 적절히 사용하는 것이 바람직하다[2,9].

4.1.2. 알고리즘

알고리즘은 특정 문제를 해결하기 위해 정해진 일련의 절차이다. 그런데 컴퓨터과학의 알고리즘 과목에서 다루는 특정한 문제는 주로 정렬, 탐색과 같은 매우 정형화된 문제들이다. 그러므로 CT가 지향하는 일상생활 속에서 혹은 각 전공 속에서 발생하는 비정형적인 문제를 다루지 않기 때문에 알고리즘을 깊숙이 교육시키는 것은 CT 역량 함양에 투자 대비 별 도움이 되지 않는다.

4.1.3. 자료구조

자료구조는 컴퓨터에서 처리할 자료를 효율적으로 관리하고 구조화시키기 위한 것으로 스택, 큐, 배열, 리스트 등을 다룬다. 그런데 이 영역은 CT 교육 차원

에서는 JAVA 에서 제공하는 Collection Framework 수준으로 그 개념과 사용법에 국한시킨 교육만으로도 충분해 보인다.

4.1.4. 소프트웨어공학

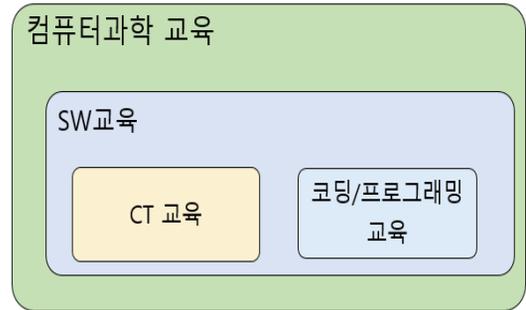
소프트웨어공학은 소프트웨어 개발을 위해 계획, 분석, 설계, 구현, 테스트, 유지보수 등의 일련의 과정을 체계적으로 다룬다. 그런데 여기서 분석과 설계 과정에 포함된 다양한 개념들이 (표 1)의 K-12에서 다루는 CT 요소와 가장 근접하다. 그러나 불행히도 이들 각 요소들이 비전공자들도 다양한 실습과정을 통해 확실히 습득할 수 있을 만큼 체계적인 연구와 교육 자료가 구비되어 있지 않다는 아쉬움이 있다.

지금까지 살펴본 것처럼 CT 개념에 바로 부합되는 혹은 CT 교육을 위해 직접적으로 사용될 수 있는 컴퓨터과학의 단일 과목은 존재하지 않는다. 따라서 미국에서는 CT 교육을 위해 컴퓨터과학에 있는 교과과정을 수정하여, 단지 프로그래밍이 아니라 CT를 위한 근본적인 원리와 개념들을 포함시키려는 시도가 행해지고 있다[10].

4.2. 용어간 관계 및 교육 방향 정립

많은 사람들이 CT에 대해 생소하게 느끼거나, 알고 있더라도 잘못 인식하고 있는 경우가 많다. 특히 CT 교육을 단지 코딩 교육의 일종으로 잘못 인식하고 있거나, CT가 문제해결력 함양이 궁극적인 목적임을 알고 있음에도 컴퓨터과학에서조차 문제해결력 함양을 위한 구체적인 과목이 정립되어 있지 않기 때문에 앞 절에서 언급한 과목 중에서 가장 가르치기 용이하고, 시각적인 결과물이 보이는 프로그래밍 교육에 집착하는 경향이 있다.

(그림 2)에서 미래부는 SW 교육은 ‘창의적 아이디어를 SW로 구현하는 사고력 교육’이며, 초중등 학생들에게 SW 교육이란 쉬운 것이고, 스스로 즐겁게 익힐 수 있고, 몰입과정에서 저절로 심화되는 교육이 되어야 한다고 정리하였다. 그리고 SW중심대학사업에서는 CT를 ‘SW의 가치와 중요성을 이해하고, 컴퓨터과학의 원리를 활용하여 문제를 분석·해결하는 역량’



(그림 4) SW 교육의 개념 구조도

으로 정의하였다[9].

이를 종합하면 각 용어의 개념 관계는 (그림 3)과 같이 도해할 수 있다. 즉 초중등교육에서의 SW 교육이란 CT 교육에 빙점을 두고, 코딩/프로그래밍은 CT를 효율적으로 흥미 있게 습득시키기 위한 일종의 도구이다. 그리고 대학에서 SW 교육의 목표는 SW 전공자에게는 기업이 원하는 현장 맞춤형 인재 양성, 그리고 비전공자에게는 자신의 전공과 더불어 SW 소양을 겸비한 융합인재 양성으로 정리된다. 결국 초중등교육이나 대학교육에서 범용적인 SW 교육은 ‘컴퓨터과학에 입각한 문제해결력 함양’ 즉 CT 역량 함양에 두고 있다고 할 수 있다.

초중등 학생들이나 비전공자를 위한 SW 교육의 목표는 전공자를 대상으로 한 교육과는 당연히 달라야 한다. 전공자를 대상으로 한 SW 교육의 목표는 SW를 개발하는 개발자나 설계자를 육성하여, SW 관련 분야에 종사할 인재를 배출시키는 것이다. 그러나 초중등 학생이나 비전공자를 대상으로 한 SW 교육의 목표는 코딩에 빙점이 있지 않고, 미래 인재로서 필요한 역량을 길러 주는 것이므로, 학습자의 사고력과 창의성, 문제해결력을 배양시키는 CT 교육에 역점을 두어야 한다.

5. 결 론

CT는 디지털 세대에서 모든 사람이 갖추어야 할 기본적 역량으로서 이는 단순한 ‘컴퓨터 활용 능력’이나 ‘코딩 능력’과는 차원이 다른 창의적 문제해결 능력이다. CT는 21세기가 요구하는 핵심 역량 중 하나

로 주목받고 있으며 새로운 디지털 시대에 모든 학문의 기본 소양으로서 그 중요성이 크게 강조되고 있다.

지금까지 수학과 영어를 당연한 국민적 소양으로 인식하며 누구나 배워왔다. 그리고 수학을 배우면서 수학자 양성 교육으로 생각하지 않았고, 영어를 배우면서 영어 번역사나 통역사 양성 교육으로 받아들이지 않았다. 그런데 CT 교육에 대해서는 유독 SW 개발자, 프로그래머 육성 교육으로 보아 거부감을 느끼는 경우도 상당하다.

그러나 미국의 컴퓨터과학 교사협회(CSTA)에서 발표한 컴퓨터과학 교육과정에는 단계별로 CT 역량을 길러주는 목표에 초점을 맞추고 있고[6], 정부가 추진하고 있는 범용적인 SW 교육은 CT 역량 함양에 비중을 두고 있다는 점을 상기해야 한다.

CT는 ‘문제해결을 위한 전산적/컴퓨터과학적 사고’라는 틀 속에서 이를 구성하는 세부개념 혹은 구성요소들은 현재 계속 진화과정에 있다. 따라서 앞으로 컴퓨터과학과 컴퓨터교육 분야에서 CT에 대한 체계적인 구성요소와 구체적인 교육 방법론을 수립하려는 연구가 지속적으로 진행될 필요가 있다.

참 고 문 헌

- [1] 오바마, <http://www.software.kr/um/um04/um0401/um040102/um040102View.do?postId=15589>, 2016.
- [2] 최현숙, SW 교육에 빠진 기관·기업들: ‘쉽고 재밌는 SW’ 화두로 미래세대 공략, 머니투데이 테크, <http://www.mt.co.kr/view/mtview.php?type=1&no=2015112518035647748&outlink=1>, 2015.
- [3] Jeanette M Wing, “Computational thinking”, *Communications of the ACM*, Vol. 49, No 3: 33, 2006.
- [4] Cuny, J. Synder, L. & Wing, J. M. *Demystifying Computational Thinking for Non-Computer Scientists*, 2010.
- [5] 최숙영, “21st Century Skills와 Computational Thinking 관점에서의 ‘정보’ 교육과정 분석”, 한국컴퓨터교육학회 논문지 제14권 제6호, 2011.
- [6] CSTA, *CSTA K-12 Computer Science Standards*, <http://csta.acm.org/Curriculum/sub/K12Standards.htm>, 2015.
- [7] 김수환, “Computational Thinking 교육에서 나타난 컴퓨터 비전공 학습자들의 어려움 분석”, 한국컴퓨터교육학회 논문지 제18권 제3호, 2015.
- [8] 한국컴퓨터교육학회, 정보통신산업진흥원, *국내의 SW 교육 운영현황 및 요구사항 조사보고서*, 2014.
- [9] 교육부·미래부, *초등학교에서 대학까지, 소프트웨어(SW) 교육 청사진 나왔다 !, 「SW중심사회를 위한 인재양성 추진계획」* 발표, 2015.
- [10] Jeanette M Wing, *Computational thinking—What and Why?.* CMU Research notebook. <http://link.cs.cmu.edu/article.php?a=600>, 2011.

저 자 소 개



양 단 희

1989년 연세대학교 전산학과(이학사)
 1991년 연세대학교 대학원 전산학과(이학석사)
 1999년 연세대학교 대학원 컴퓨터학과(공학박사)
 1991년~1995년 현대전자 S/W 연구소
 2013년 Visiting Scholar at Texas A&M University
 2001년 3월~현재 평택대학교 컴퓨터학과 교수
 관심분야: 멀티미디어, 컴퓨터보안, 기계학습, 소프트웨어공학, 컴퓨터교육