

# 컴퓨터 코오스웨어의 개발절차와 평가 준거

柳 完 永  
(서울女大 教育心理學科)

컴퓨터 코오스웨어란 컴퓨터를 통하여 敎科를 학생들에게 가르칠 목적으로 제작된 敎授用 소프트웨어 혹은 敎授用 프로그램을 뜻한다. 이 글에서는 코오스웨어의 개발 절차와 개발된 코오스웨어를 평가·선별하는 데 고려해야 할 準據들을 제시하려 한다.

## I. 머리말

컴퓨터 코오스웨어(Courseware)란 컴퓨터를 통하여 敎科를 학생들에게 가르칠 목적으로 제작된 敎授用 소프트웨어 혹은 敎授用 프로그램을 가리킨다. 코오스웨어에 대한 관심은 컴퓨터가 최근에 敎育現場에 많이 보급됨에 따라 이를 교실 수업에 어떻게 효과적으로 이용할 것이냐에서 비롯되었다고 말할 수 있다. 이는 컴퓨터를 하나의 敎育工學의 媒體로서 활용하여 수업의 質과 效果를 높이려는 노력이라고도 해석할 수 있다.

敎育工學의 측면에서 컴퓨터는 TV나 슬라이드 등과 같은 교수매체와는 좀 다른 특성을 갖고 있다. 그 특성들이란 사용자와 컴퓨터, 컴퓨터와 컴퓨터끼리 얼마든지 交互作用(interaction)이 가능하고, 프로그램의 작성에 따라 학생 개인의 準備도와 학습속도 등을 감안한 個別

化授業이 가능하다는 점이며, 주컴퓨터(master computer)와 부속컴퓨터(slave computer)를 연결하여 사용할 때 교사는 주컴퓨터를 통하여 부속컴퓨터를 사용하고 있는 학생들을 얼마든지 통제할 수 있다는 이점이 있다. 그러면서도 컴퓨터는 TV처럼 소리, 색채, 그래픽 등을 나타낼 수 있는 기능들을 다 할 수 있다. 그러나 컴퓨터가 아무리 좋은 교육공학적 기능을 갖추고 있다 할지라도 모니터에 재현되는 모든 것은 코오스웨어에 달려 있기 때문에 코오스웨어의 중요성은 언급할 필요조차 없다.

전통적인 敎授方法보다 컴퓨터를 통한 敎授에 학생들이 훨씬 더 호감을 보인다는 점, 학습시간을 단축시켜 준다는 점, 컴퓨터 활용에 대한 긍정적 태도를 길러 준다는 점, 그리고 학습적성이 낮은 학생들에게 특히 높은 학습효과를 보인다는 점 등이 발견되었다(Ragsdale, 1982). 그러나 이와 같은 효과에도 불구하고 1950년대말 처음 그 개념—이를 들면 CAI(Computer Assisted

Instruction)나 CBI(Computer Based Instruction) 등이 창안되었던 때의 기대만큼 큰 진전을 보지 못하고 있는 게 현실이다. 그 이유는 여러 가지가 있겠지만 무엇보다도 컴퓨터 하드웨어의 미진한 보급, 컴퓨터를 사용할 줄 아는 敎育者의 不在, 그리고 良質의 코오스웨어의 稀貴를 들 수 있다. 근래에 와서는 마이크로 컴퓨터의 발달에 따라 성능 좋고, 저렴하며, 사용하기 간편하고, 나아가 외형적 규모가 작은 컴퓨터가 學校 現場에 널리 일반화됨에 따라 CAI의 성패 여부는 良質의 코오스웨어를 많이 개발해 낼 수 있느냐 없느냐에 달려 있다고 많은 전문가들은 보고 있다.

미국의 경우 대부분의 코오스웨어들은 프로그램 언어를 아는 교사나 컴퓨터 애호가, 또는 컴퓨터 회사에 의해 개발되고 있다. 이들의 의욕을 저하시켜서는 안 되겠지만 현재 시중에 나와 있는 대부분의 코오스웨어들은 질적으로 픽 낮게 평가되고 있다(Galanter, 1983). 이 같은 코오스웨어를 수업현장에 투입하여 본 교사들 사이에는 컴퓨터의 敎授的 活用に 대해 벌써부터 회의감을 갖고 있는 사람도 있다. 이제 막 교실 수업에 컴퓨터를 적용해 보려는 우리로서는 양질의 코오스웨어를 개발할 수 있는 방안이 절실히 요구된다 하겠다.

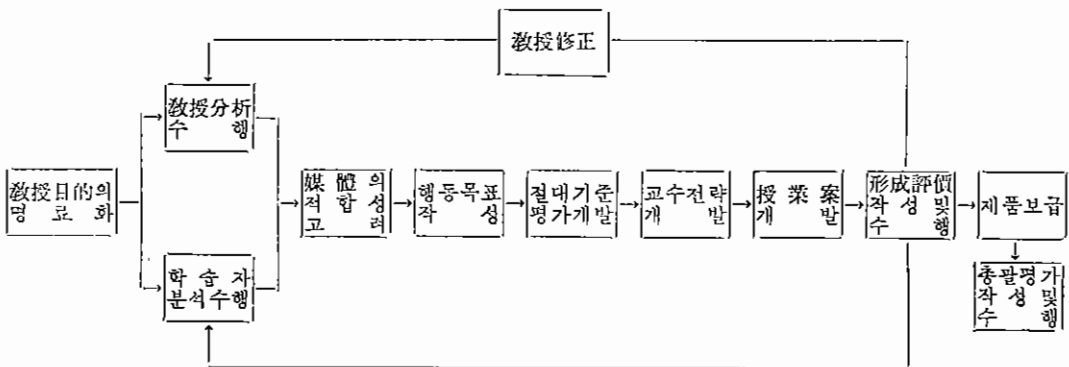
코오스웨어를 개발하기 위해서는 敎授設計의 인 측면에서 일반적으로 거쳐야 할 단계가 있고, 코오스웨어들이 갖추어야 할 일반적인 요건들이 있다. 따라서 본 글에서는 코오스웨어의 개발

절차와 개발된 코오스웨어를 평가·선별하는 데 고려해야 할 준거들을 제시하려 한다.

## II. 코오스웨어의 開發절차

코오스웨어는 수업의 過程을 컴퓨터의 특성을 살려 프로그램화한 것이기 때문에 一般的인 敎授·學習資料의 開發過程과 큰 差異가 없다. 예를 들어 우리가 한 단위의 수업을 도입, 전개, 정리, 평가의 순서로 진행하려 한다면 이에 맞게 學習指導案을 작성하여, 컴퓨터 언어로 프로그램화하는 작업이 뒤따르게 될 것이며, Glaser의 수업모형에서처럼 수업의 흐름을 ‘授業目標’, ‘投入行動’, ‘授業節次’, ‘學習評價’로 잡는다면 이 이론에 맞는 수업자료를 開發한 후, 이를 컴퓨터 언어로 記號化하게 된다. 지금까지 敎授資料의 開發過程과 프로그램화 作業을 結合하여 만든 코오스웨어의 개발절차 모형은 많은 사람들에 의해 다양하게 묘사되고 있다. 이들의 諸模倣들이 가지고 있는 공통적 특성은 체계적이고도 계획된 과정을 거치게 함으로써 敎授의 效果와 효율성을 높이려 한다는 점이며, 다른 한 가지는 目標, 학습활동, 그리고 학습결과의 査定간에 一致性을 한결같이 강조하고 있다는 점이다.

여기서는 Dick & Carey(1978)와 Gagné & Briggs(1974)에 터하여 코오스웨어 개발모형을 제시하고, 각 단계별 주요 활동을 간략히 언급하려 한다.



<표 1> 코오스웨어 개발절차 모형\*

\* Dick & Carey (1978)를 인용한 Smith & Boyce (1984)에서 재인용함.

앞의 <표 1>에서 보는 바와 같이 코오스웨어를設計할 때에는 코오스웨어를 통하여 가르치려는 一般의 目的을 분명히 밝히고, 그 목적을 教授分析을 통하여 系列的인 學習段階로 세분화하며, 學習者의 諸特性을 記述하는 과정들이 선행되어야 한다. 특히 학습자의 특성을 記述할 때에는 예견되는 사용자들의 컴퓨터에 관한 지식—예를 들면 키보드에 대한 숙달도, 컴퓨터에 관한 이해 수준, 불안도 등—을 포함시키지 않으면 안 된다.

다음 단계에서는 학습과제가 과연 컴퓨터라는 매체를 통하여 전달하는 것이 적절한지의 여부를 판별해야 한다. 학습과제에 따라서는 컴퓨터가 理想的으로 적합할는지 모르지만 모든 학습과제에 적절한 매체라고는 할 수 없다. 왜냐하면 코오스웨어를 통한 表現能力과 判斷能力에 한계가 있기 때문이다. 우리의 주변에서는 흔히 모든 敎科內容을 코오스웨어화하려는 경향이 있는데 이는 재고되어야 할 것이다.

컴퓨터가 教授媒體로서 적절하다는 판단이 서면 目的을 세분화하여 행동목표<sup>1)</sup>로 진술하고, 이들 행동목표에 근거하여 절대기준 평가문항을 개발하게 된다. 행동목표의 진술문에는 수업을 통해서 성취하려는 도착점 행동성표(terminal performance)와 이 행동성표가 발생되어야 할 중요 조건 또는 장면, 그리고 이 행동성표가 성공적인지 아닌지를 판단하기 위한 수락 기준이 포함되어야 한다고 Mager는 주장한다. 앞에서 언급한 것처럼 절대기준 평가문항은 이들 행동목표들을 성취했는지의 여부를 판별하는 방향으로 제작되어야 한다.

코오스웨어에서 사용될 수 있는 평가문항의 類型은 컴퓨터에 프로그램화되어 사용자의 반응을 판단할 수 있는 컴퓨터의 능력에 한계가 있기 때문에 한정받다고 말할 수 있다. 다시 말하면 논문형과 같은 자유반응식의 문항은 컴퓨터가 파악하기 어려우므로 곤란하다. 코오스웨어에서 흔히 사용되는 문항의 유형은 眞僞型,

選多型, 短答型, 完結型, 連結型, 그리고 行動型 등이 있다.<sup>2)</sup> 이때 문항의 類型의 선택은 학습을 통하여 성취하려는 학습목표에 준하여 이루어져야 한다.

평가문항의 개발이 끝나면 教授戰略을 세워야 한다. 코오스웨어에서 사용되는 教授戰略로서는 흔히 훈련 및 연습형(drill and practice), 個人指導型(tutorial), 示範型(demonstration), 시뮬레이션형(simulation), 그리고 教授 게임型(instructional game)을 들 수 있다. 이들 전략 속에 숨어 있는 기본적인 假定과 프로그램상의 특징들은 여러 곳에서 발견할 수 있다(Kraus, 1982; Coburn et al., 1982). 특정한 教授戰略은 특정한 學習結果를 위해 여타 教授戰略보다 더욱 적합할 수 있으므로 교수전략의 선정시에는 특히 성취하려는 행동목표에 특히 유의해야 한다. 각기의 행동목표를 성취하는 데는 꼭 한 가지의 전략을 사용해야 된다는 법칙은 없으므로 둘 또는 그 이상의 전략을 조합하여 짤 수도 있는 것이다.

학습전략이 일단 선정이 되면 單位別 授業案을 짜게 된다. 여기서의 授業案이라고 하면 수업사태에서 일어날 수 있는 사건들을 체계적으로 흐름에 따라 정리해 놓는 것을 뜻한다. 수업안의 작성요령에 관해서는 사람마다 다를 수 있을 것이다. 여기서는 학습에 필요한 內在的 認知過程에 맞춰 수업사태의 흐름을 系列化해 놓은 Gagné와 Briggs(1974)의 案을 제시하고자 한다.

#### 1) 注意획득

이의 목적은 학습자들의 관심을 학습과제에 초점을 맞추도록 자극하는 데 있다.

#### 2) 학습자에게 目的을 알리기

학습자에게 目的을 분명히 밝혀 주는 이유는 수업을 통하여 무엇을 가르치려는지 수수께끼로 남겨 헛되이 학습자의 정력을 낭비할 필요가 없다는 데 있으며, 오히려 認知的 정력을 특정한 목표에 맞추므로써 기대감을 확립하고, 학습 발

1) 교수목표의 진술과 행동목표에 관해서는 김호련의 「現代教授理論」(敎育出版社, 서울, 1976) pp. 38~50을 참조할 것.

2) 검사문항의 유형과 작성기술에 관해서는 任寅宰의 「絕對基準評價의 原理와 實際」(베영사, 서울, 1976), pp. 151~182를 참조할 것.

생시 이를 깨달도록 하는 데 있다.

### 3) 先須技能的 再生을 자극하기

이 단계에서는 새로운 목표를 학습하기 위해서 앞서 배운 능력이나 지식 중 도움이 되거나 필요한 것들을 돌이켜 보도록 자극하는 것이다.

### 4) 본시 학습자료의 제시

이 시점에서 학습자가 학습하게 될 자료가 처음으로 드러내 놓여진다. 이때의 학습자료는 문장, 그래픽, 청각, 혹은 시각적 형태로 표현된다.

### 5) 학습안내 제공

이 단계의 목적은 나타난 학습자료 중 木質의인 부분에 학습자의 주의를 쏠도록 안내해 줌으로써 학습의 효과를 높여려는 데 있다. 학습안내를 위한 技法으로서는 단서, 암시, 조언, 직접적인 도움 등이 쓰인다.

### 6) 행동 유도

여기서의 행동 유도란 학습자가 학습한 것을 演示해 보일 수 있도록 기회를 제공하는 것을 뜻한다. 행동 유도를 하기 위해서는 질문이 있어야 한다.

### 7) 피이드백(feedback)의 제공

피이드백의 목적은 앞의 수업상태에서 학습자가 보인 행동의 적절성에 관하여 학생들에게 알려 주는 데 있다. 컴퓨터는 個別的이면서도 즉시적인 피이드백 제공을 위한 가장 효과적인 도구로 알려져 있다. 컴퓨터에 의한 피이드백의 장점들은 질문의 제시와 학습자 반응 사이를 기다릴 수 있다는 점, 학습자가 예정된 시간 동안 반응하지 않을 경우 질문을 다른 방식으로 재진술해 주거나 도움 문장을 줄 수 있다는 점, 그리고 誤答을 반응할 때마다 좀 더 직접적인 피이드백을 제공함으로써 한 문항에 대해 여러 번 답변을 시도하게 할 수 있다는 점이다.

### 8) 成就의 査定

수업의 흐름에서 마지막 단계 중의 하나는 학습결과의 査定이다. 학습결과의 사정 혹은 教授效果의 評價를 體系的으로 이루기 위해서는 문항의 신뢰도와 타당도를 높여야 한다. 成就의 査定과 관련하여 컴퓨터가 가지고 있는 두 가지 독특한 특성을 든다면 미리 제작된 문항 풀(pool)로부터 우선적으로 문항을 선정할 수 있

다는 점과 학습자의 반응(응답)에 따라서 다양한 수자의 문항을 제시할 수 있다는 점이다. 이것은 학습자의 반응에 따라서 프로그램의 수행 중에 검사의 구성을 변경할 수 있도록 개발해야 됨을 시사한다고 볼 수 있다.

### 9) 把持와 傳移의 제고

코오스웨어를 통한 학습에서는 흔히 포함되지 않지만 이 단계에서는 학습자에게 복습시키거나 知識을 정련시킴으로써 앞으로의 응용이나 사용을 원활하게 하는 데 목적이 있다. 파지와 전이의 효과를 높이기 위한 방법으로서 학습자가 잘못 반응한 질문을 다시 해보게 하거나, 다르게 진술된 질문들에 응답하게 함으로써 복습하게 하는 방법, 이에 학습한 概念이나 原理를 검토할 수 있도록 많은 例示的 資料를 제공해주는 방법 등이 쓰인다.

지금까지 Gagné와 Briggs의 이론에 따라 특정한 수업에서 일어날 수 있는 수업상태의 흐름을 계열화하여 제시하였다. 그러나 모든 수업이 이들 상태(events)를 모두 포함해야 한다거나, 꼭 이 순서대로 진행되어야 한다고는 말할 수 없다. 하지만 이것들은 가장 있을 법한 순서대로 정리된 것이다(Gagné & Briggs, 1974).

수업안이 일단 작성이 되면, 이것을 특정한 컴퓨터 언어를 사용하여 프로그램화하게 된다. 教授의 主要事態別 코오스웨어 개발상의 주요 技法들은 Smith와 Boyce(1984)에 의해 잘 정리되어 있다.

앞의 <표 1>에서 보는 바와 같이 授業案開發과 프로그램 짜는 作業이 끝나면 形成評價를 設計하여 이를 실시하게 된다. 여기서의 형성평가란 한마디로 코오스웨어를 통한 수업의 査定과 修正의 段階를 나타낸다. 이전 단계까지는 학습이 이루어질 것이라는 가정하에서 진행되어 왔는데 형성평가에서는 실제로 이 가정을 검증하고 수정이 필요한 부분은 수정하게 된다. 여기서의 修正이란 컴퓨터 프로그램을 다시 짜는 것까지를 포함한다.

形成評價에서는 標的集團의 標集으로부터의 資料수집이 강조된다. 코오스웨어의 형성평가를 위한 자료수집의 방법으로 Golas(1983)는 첫째 컴퓨터 프로그래밍을 하기 이전에 종이에 쓰여

진 수업안을 가지고 최소한도 한 학생에게 시험을 해보고, 둘째 컴퓨터 프로그램화된 수업을 통하여 한 학생에게 학습을 하도록 해보며, 셋째 15명 내외의 소집단 학생들에게 컴퓨터 프로그램화된 수업을 실시해 보고, 최종적으로 현장 검사를 실시할 것을 주장하고 있다. 이상의 각종 現場實驗時에는 內容側面, 學習理論的 側面, 프로그램 作動側面, 學生 學習側面 등에 관심을 가지고 관찰을 해야 되는데, 각 側面別로 고려되어야 할 세부사항에 관해서는 다음 절에서 자세히 論議하겠다.

코오스웨어 개발의 마지막 단계는 생산품을 보급하는 일이다. 코오스웨어를 통한 수업의 시행을 증진시키기 위해서는 개발된 특정한 코오스웨어가 기존 教育課程에 어떻게 統合될 수 있는지에 관하여 합당한 文獻와 提議를 포함시켜야 한다.

### Ⅲ. 코오스웨어의 評價準據

#### 1. 諸理論들

코오스웨어를 開發할 때에나—특히 앞에서의 形成評價 단계에서—이미 開發된 코오스웨어를 평가할 때에는 고려되어야 할 몇 가지 준거들이 있다. 이는 사용상의 어려움이나 학습을 저해하는 요소들을 教授設計나 컴퓨터 프로그램 작성 과정에서 막을 수 있고, 모니터를 통한 再現을 보다 효과적으로 함으로써 학습의 효율을 높일려는 데 그 意義가 있다.

코오스웨어의 평가 준거들을 Roblyer(1981)는 本質的 特性, 審美的 特性, 分化的 特性으로 分類하여 제시하고 있으며, Jay(1983)는 認知的 接近이라는 측면에서 기억 및 주의, 언어와 문장특성, 도표와 시각과정, 학습자의 認知的 特性, 그리고 피이드백 등으로 나누어 평가의 基準을 제시하고 있다. 또한 Wager(1981)는 技術上的 質, 內容의 正確性, 教授上的 質, 學習者의 類型 등의 諸側面으로 나누어 評價準據들을 記述하고 있다. 한편 Cohen(1983)은 마이크로 컴퓨터 코오스웨어의 設計時에 고려되어야 할 필수적 특성들을 ‘教授設計에 관한 事項’과 ‘코오스웨어 設計에 필요한 事項’으로 區分하여 자

세히 진술하고 있다.

앞에서 살펴본 것처럼 이들이 나눈 범주들은 각기의 관점에 따라 다르지만 각 범주 속에 들어 있는 세부 준거들은 상호 유사한 점이 많다. 이들 세부 준거들을 Ragsdale(1982)의 分類基準에 따라 內容側面, 學習理論側面, 프로그램 作動側面, 그리고 學習者側面으로 整理하여 제시하면 아래와 같다.

#### 2. 評價의 準據

##### 〈내용 측면〉

1) 내용이 대상학생 집단에 적합한가? 특히 사용 용어가 대상학생에 적합한지의 여부, 대상 집단의 明示 여부, 準備度에 대한 言及 여부에 관심을 두어야 한다.

2) 코오스웨어에 담긴 내용이 教育課程의 目的에 合致하는가?

3) 내용이 정확한가?

4) 목적이나 목표가 직접적으로든 간접적으로든 분명히 제시되었는가? 목표의 진술과 관련하여 細分化 여부, 행동용어의 사용 여부, 학습자 중심 진술 여부 등에 주의를 기울여야 할 것이다.

##### 〈학습이론적 측면〉

1) 피이드백이 적절히 사용되었는가? 피이드백이 위협적이거나 비꼬는 식이 되어서는 안 된다는 점, 正·誤答의 여부를 학습자에게 즉시 알려 주어야 된다는 점, 피이드백이 불필요한 과잉강화가 되어서는 안 된다는 점, 피이드백이 오히려 오답을 강화하는 식이 되어서는 안 된다는 점, 그리고 단순히 “맞았어요” “틀렸어요”라는 식의 피이드백보다는 맞았으면 왜 맞았는지, 틀렸으면 왜 틀렸는지를 충분히 설명해 주고, 피이드백으로 사용되는 文句 또한 다양하면서도 私的으로 제공되어야 한다는 점 등을 고려해야 한다(Waldrop, 1984).

2) 코오스웨어에 담겨 있는 教授戰略들이 授業事態에 알맞게 결합되어 사용되었는가?

3) 학습자 개개인의 필요에 따라서 컴퓨터 프로그램을 수정 가능하도록 되어 있는가?

4) 코오스웨어는 教師의 도움 없이도 학생들이 스스로 조작하며 공부할 수 있을 만큼 完備되어

있는가?

5) 個別學習, 小集團學習 등 다양한 수업상황에서 사용할 수 있는가?

6) 학습자의 다양한 감각기능을 두들길 수 있도록 프로그램이 작성되어 있는가? 이 뜻은 컴퓨터의 색채, 소리, 도표, 수치, 그리고 언어적 기능들을 심분 활용할 수 있도록 프로그램을 짚으로써 학생들의 학습을 고양시키려는 것이다.

〈프로그램 작동 측면〉

1) 프로그램상의 오류나 작동 중간에 멈추는 일이 없는가?

2) 코오스웨어는 학습자의 잘못 반응을 처리할 수 있도록 방비되어 있는가?

3) 학습자가 프로그램의 작동을 자유로이 통제할 수 있도록 되어 있는가? 프로그램의 작성에 있어서 학습자 통제(learner control)로 할 것인가, 아니면 프로그램 통제(program control)로 할 것인가는 많은 논란의 여지가 있다(Roblyer, 1981; Cohen, 1983; Waldrop, 1984 참조). 왜냐하면 학습자가 자유로이 공부를 계속하고 싶으면 계속하고 중단하고 싶으면 중단하도록 하는 경우와, 학습자 이외의 사람이—예를 들면 敎師나 父兄가—일정한 시간 동안 공부를 해야 중단할 수 있도록 허락하는 경우에서처럼 각기의 長短點이 있기 때문이다.

여기서는 학습자가 내용에 따라서 뛰어 넘을 수도 있고, 뒤로 거슬러 올라가 복습할 수 있도록 프로그램의 통제를 벗어나야 한다는 점과 학습자가 자유로이 공부할 수 있을 때에 학습의욕이 증진될 것이라는 가정에서 학습자 통제를 선택했다.

4) 각 화면마다 학습자가 이해할 수 있도록 분명하고도 받아들일 수 있는 지시가 주어졌는가? 흔히 프로그램 시작시에 지시가 한꺼번에 주어지는데, 다음 화면으로 넘어가면 지워져 버리기 때문에 학습자가 컴퓨터에 줄 수 있는 지시(반응)를 잊어버리는 수가 있기 때문이다.

5) 학습자가 각 화면의 내용을 충분히 이해할 수 있을 만큼 충분한 시간이 주어졌는가?

6) 敎師와 학생을 위해서 코오스웨어에 관한 분명하고도 유익한 지침서가 함께 마련되어 있는가?

7) 도표나 음향 그리고 색채 등은 내용과 有關하며, 학습자의 수준에 비추어 볼 때 유치하지 않는가?

8) 화면에 나타나는 文章構成이 읽기 쉽도록 짜여져 있는가? 단어와 단어 사이 혹은 줄과 줄 사이를 충분히 띄우고, 한 화면에 한 아이디어를 제시하는 방법이 좋을 것이다.

9) 반응시간(response time)과 로딩시간(loading time)이 빠르게 프로그램되었는가?

〈학생학습 측면〉

1) 학생들이 사용하기 쉬운가?

2) 학생들에게 흥미롭게 구성되었는가?

3) 프로그램을 통해 가르치고자 하는 것을 학습자가 배웠는지 확인할 수 있는 事前·事後檢査와 같은 評價部分이 있는가?

4) 코오스웨어를 사용할 때 발생할지도 모를 비의도적인 학습상의 문제점은 없는가?

5) 컴퓨터를 통한 敎授와 전통적인 敎授 사이의 效果를 비교해 보아야 한다.

#### IV. 맺는 말

지금까지 컴퓨터를 통한 敎育에서 코오스웨어의 중요성과 良質의 코오스웨어 개발의 필요성을 논하고, 코오스웨어 개발의 一般的 節次와 그 評價準據들을 살펴보았다.

코오스웨어의 개발은 일반적으로 敎科專門家와 敎授設計者, 그리고 프로그램 作成者로 구성된 팀 접근방식을 택해야 된다고 많은 사람들은 보고 있다. 敎科專門家は 內容에 대한 知識과 學習者의 特性에 관한 知識을 가지고 기여하고, 敎授設計者는 授業準備를 위한 體制的인 계획을 제공하며, 수업의 과정에 學習의 原理를 적용함으로써 開發을 안내하고, 프로그램 작성자는 컴퓨터의 능력을 파악하여 敎科專門家와 敎授設計者의 계획을 프로그램화하는 작업을 맡게 된다. 흔히 우리 주변에서는 소요 경비를 줄이고 프로그램을 개발해 놓고 보자는 의도하에서 프로그래머 혼자서 개발해 가는 경향이 있는데 이는 특히 경계되어야 할 일이다.

한 시간 분의 코오스웨어를 개발하는 데 소요되는 총시간은 開發에 참여한 人力의 資質과 코

오스웨이의 내용 및 형태 등에 따라 달라지겠지만 약 100시간 정도라 한다(Terashita, 1983). Gleason(1981)은 우수한 한 시간 짜리 프로그램을 만드는 데 10,000달러 가량이 들 것으로 예측한다. 이 두 사람의 수치는 코오스웨어 개발이 많은 인력과 경비가 소요됨을 단적으로 나타낸다. 따라서 컴퓨터를 통한 敎育의 發展을 가져 오기 위해서는 많은 재정적 투자가 요청된다고 하겠다. \*

〈參考文獻〉

金榮權, 「現代敎授理論」, 서울: 敎育出版社, 1976.  
 류완영, 홍운선, 김선오, 「초·중등학교 컴퓨터敎育을 위한 기초연구」, 한국敎育개발원, 연구보고 RR. 83-23, 1983.  
 柳完永, 李相路, 劉銀珍, “컴퓨터敎育課程 開發研究,” 「敎育課程研究」, 敎育課程研究會, 第四輯, 1984.  
 Coburn, P., Kelman, P., Roberts, N., Snyder, F.F., Watt, D.H., & Weiner, C. *Computers in education*. Mass.: Addison-Wesley Pub. Inc., 1982.  
 Cohen, V.B. “Criteria for the evaluation of microcomputer courseware.” *Educational Technology*, Jan. 1983, 9-13.  
 Dick, W., & Carey, L. *The Systematic design of instruction*. Glenview, IL.: Scott, Foresman, 1978.  
 Gagné, R.M., & Briggs, L.J. *Principles of instructional design*. New York: McGraw-Hill, 1974.

Galanter, E. *kids and computers*. New York: A GD/Perigee Book, 1983.  
 Gleason, G.T. “Microcomputers in education.” *Educational Technology*, March 1981.  
 Golas, K.C. “The formative evaluation of Computer-assisted instruction.” *Educational Technology*, 1983, 23(1), 26-28.  
 Jay, T.B. “The Cognitive approach to Computer courseware design and evaluation.” *Educational Technology*, Jan. 1983.  
 Klaus, W. Microcomputers in education. In D.L. Zalewski(ed.), *Microcomputers for teachers with application to mathematics and science*, The School Science and Mathematics Association, Inc., 1982, 33-42.  
 Robliyer, M.D. “Where is it good courseware? Problems in developing standards for microcomputer courseware.” *Educational Technology*, Oct. 1981.  
 Smith, P.L., & Boyce, B.A. “Instructional design considerations in the development of computer-assisted instruction.” *Educational Technology*, July 1984, 5-11.  
 Terashita, Y. *CAI: Software development and implementation in engineering education*. Paper presented at the UNESCO-AEU Workshop on Computer Assisted Instruction, Jakarta, Indonesia, Oct. 1983.  
 Wager, W. “Issues in the evaluation of instructional computing programs.” *Educational Computer*, Sep./Oct. 1981, 1(3).  
 Waldrop, P.B. “Behavior reinforcement strategies for computer-assisted instruction: Programming for success.” *Educational Technology*, Sep. 1984, 38-41.