

# SW 적응력 향상을 위한 SW 교수·학습 전략

유인환·구덕희  
대구교육대학교 전산교육과

## 요 약

최근 ICT 교육의 활성화에 따라 응용 SW(Software) 교육의 중요성이 더해지고 있다. 그런데 지식의 기하급수적인 팽창과 컴퓨터 기술의 급속한 발전에 따라 끊임없이 새로운 SW가 양산되고 있기 때문에 기존의 SW 학습 패러다임은 여러 가지 문제에 봉착하였다. 본 연구는 이와 같은 문제의식을 가지고 문제 상황에서 학습자 스스로 SW의 적절한 기능을 쉽게 찾아 학습하고, 이를 문제 사태에 적용, 효율적으로 해결할 수 있으며, 과지와 전이를 촉진시킬 수 있는 능력을 SW 적응력이라 정의하고, 이를 배양할 수 있는 SW 교수·학습 방법을 탐색하였다. 이를 위해 SW 사용성과 UI(User Interface)디자인을 위한 원칙 등을 살펴보고, SW 적응력, UI 디자인 원칙에 따른 SW 학습 전략을 도출하였다. 또한 시범·실습식 SW 교수·학습 방법의 문제점을 살펴보고 SW 적응력 향상을 위한 탐구적 교수·학습 모형을 제안하였다.

## Software Teaching · Learning Strategy for Improvement of Software Adaptability

In-Hwan Yoo · Duk-Hoi Koo

Daegu National University of Education, Dept. of Computer Education\*

### ABSTRACT

The activation of the latest ICT(Information and Communication Technology)education has been putting more importance on the education of the application SW(software). By the way, the geometric progression of knowledge and the fast development of computer technology have continuously created new SW. Accordingly, the previous traditional SW learning paradigm runs into various problems. This study starts from the recognition of such problems. In this study, the SW adaptability is defined as the ability that learners can efficiently find and apply the suitable functions of SW to those problems in the problematic circumstances as well as promote retention and metastasis. With these premises, this paper makes an inquiry into the teaching-learning method. Besides, This paper explores the SW usability and the principles of UI(User Interface) design and deduces the SW learning strategy. Furthermore, it probes problems of the SW teaching-learning of the demonstration practice method and proposes the inquiring master for SW adaptability teaching-learning model.

**Keyword** : Computer Education, Software Teaching, Software Adaptability

### 1. 서론

지식정보사회의 시대적 요청과 이에 부응하기 위한 교육인적자원부의 정책과 각종 지원으로 현장에서 컴퓨터교육은 매우 강조되고 있는 실정이다. 그러나 현장 교육에서 컴퓨터교육은 여러 가지 어려움을

겪고 있는데 그 중의 하나가 컴퓨터 교수·학습 방법의 부족에 있다.

컴퓨터교육의 중요성과 필요성은 충분히 인식되고 있음에도 교육과정에서의 방법론 결여로 컴퓨터교육이 실효를 거두고 있지 못함은 시급히 해결되어야 할 문

제이며, 컴퓨터교육을 위한 교육방법, 교수·학습의 방법의 필요성은 다음과 같이 제기될 수 있다[10].

첫째, 다른 학습 영역과 마찬가지로, 컴퓨터교육의 대상자인 학습자의 특성에 알맞은 교수·학습 방법이 고안되어야 하며, 둘째, 컴퓨터교육에서 다루어지는 컴퓨터관련 학습 내용의 특성이 고려된 교수·학습 방법의 고안이 필요하며, 셋째, 컴퓨터교육이 이루어지는 교육 환경이 고려된 교수·학습 방법이 고안되어야 한다.

이런 관점에서 교수·학습 방법은 교육 대상, 내용, 환경에 따라 매우 다양할 수밖에 없으며, 상황적 요인에 따라 융통성 있게 처방적 대안을 제시해야 하는 경우도 있을 것이다. 따라서 보편 타당한 컴퓨터교육 방법이란 존재하기 어렵다고 볼 수 있으며, 컴퓨터교육에서 비중이 높은 영역에 대한 방법론 탐색이 우선적으로 이루어지는 것이 바람직 할 것이다.

컴퓨터 교육과정의 모형은 인식 모형, 프로그래밍 모형, 응용 SW(Software) 모형, 멀티미디어 모형, 문제해결 모형 순으로 변천되어 전개되어 오고 있다. 현재는 다양한 응용 SW 패키지, 멀티미디어, 인터넷 등을 포함하는 ICT를 활용하여, 교수·학습 과정 및 일상생활의 다양한 과제를 해결할 수 있는 능력과 정보 처리 능력 즉, 정보 접근, 분석, 종합, 적용, 평가 등을 강조하는 추세다[8].

모형에 따라 각기 목표와 내용이 달랐지만, SW에 대한 학습은 대체로 비중이 높으며, 7차 교육과정과 정보통신기술교육 지침을 분석해보면 내용에서 SW에 대한 학습이 차지하는 비중은 매우 높으므로 이를 위한 교수·학습 모형은 더욱 더 절실하다고 할 수 있다.

## 2. SW 학습에 대한 새로운 관점

### 2.1. 새로운 SW 학습 접근의 필요성

SW는 끊임없이 새롭게 개발, 업그레이드되고 있다. 이러한 상황에서 단순히 현재 많이 사용하고 중요도가 높은 SW를 선별하여 그 기능을 가르치는 방안은 근본적으로 한계를 가지고 있을 수밖에 없다. 배우고 가르쳐야 할 SW와 학습량은 계속 늘어날 수밖에 없기 때문이다. 따라서, 전통적인 학습 방법에 의존하기보다는 응용 SW의 특성에 기초한 새로운 학습 패러다임을 강구할 필요가 있다.

인간과 컴퓨터는 인터페이스를 통해 서로 대화할 수 있다. 여기에서 대화란 인간과 컴퓨터 사이의 Interaction으로써 '상호작용'이라 할 수 있다. 이를 우리는 HCI(Human Computer Interaction)이라고 한다. Bill Morgride는 Interaction Design을 처음으로

'컴퓨터 SW의 사용성에 관한 연구'에서 새로운 영역으로 개척하였다. 이로부터 사용자 인터페이스 디자인이란 개념이 탄생하였으며 HCI를 디자인 대상으로 봄으로써 인간의 지각과정과 인지과정을 그 핵심으로 두게 된 것이다. 여기서 사용자와 컴퓨터 사이의 상호 교환의 현상을 인간의 지각적, 인지적 특질로 인식하고 이를 체계화시켜서 사용자가 쉽게 대할 수 있는 인터페이스를 만들어 가는 디자인 접근법이라 할 수 있다[5].

이와 같은 SW 설계의 관점으로 SW 학습을 접근해 볼 필요가 있다. SW 개발자가 사용자를 고려하여 SW를 설계하였다면, 반대로 사용자는 개발자가 어떻게 SW를 설계하였는지에 초점을 맞추어 학습을 하면, 새로운 SW 환경에 보다 쉽게 적응하는 능력을 배양할 수 있을 것이라는 아이디어이다. 즉, SW의 설계와 기능 구성 원리에 따른 학습 전략이다.

관련 선행연구로서 Olfman과 Shayo[16]는 새로운 것이지만 예전에 사용하던 것과 비슷한 SW 학습에 대한 연구를 했다. 이들은 최종 사용자들의 SW 패키지에 대한 인지적 모형을 훈련에 의한 새로운 매핑으로 그 학습 효율을 증진시킬 수 있는 방안을 제시하고 있다.

또한 Wiedenbeck와 Zila[17]는 탐구 중심 실습 방법이 SW 학습에 효과적인 것인가라는 문제와 탐구 연습과 훈련에 최소한의 컴퓨터 배경 지식이 필요한 것인가라는 문제를 실험 연구하였다.

그러나 본 연구와 같이 SW 사용성의 관점으로 SW 학습을 접근한 연구는 찾기 힘들었다.

### 2.2. SW 사용성(usability)

인간과 컴퓨터 상호작용(interface)의 주요 개념으로 사용되는 사용성은 컴퓨터 SW 분야에서 비교적 새로운 연구 분야로서 컴퓨터 시스템을 배우기 쉽고 사용자 중심 설계과정(user-centered design process)을 통하여 사용하기 쉽게 만드는 것과 관련된다[7].

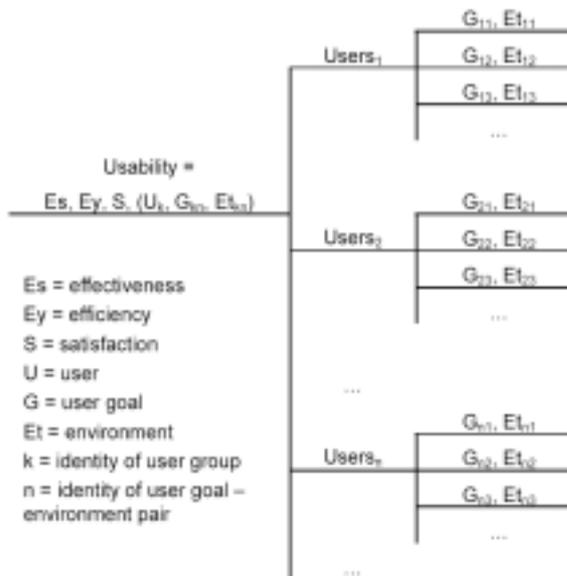
HCI 연구자들이 제시하는 응용 SW를 위한 사용성의 원리는 서로 상이하다. 일부는 인지 이론에 기초한 원리를, 또 다른 연구자들은 개인적 경험 등에 기초해서 서로 다른 원리를 제시하고 있다[13].

사용성이란 제품이나 시스템에 대한 사용자 상호작용 경험 정도를 측정할 것이라 할 수 있으며 이는 인터넷 웹 사이트, SW의 어플리케이션, 모바일, 뉴미디어 등 사용자를 필요로 하는 모든 것의 매우 중요한 부분이다. 사용성은 제품이나 시스템에 대한 사용자들의 경험에 영향을 주는 많은 요소들에 대한 결합이며, 다음의 사항들이 포함된다[18].

- 학습에 대한 용이성(ease of learning): UI(User Interface)를 경험하지 못한 사용자가 기본적인 태스크를 완성하기 위해 시스템을 얼마나 빨리 배울 수 있는가?
- 사용에 대한 효율성(efficiency of use): 시스템 사용에 대해 경험한 사용자가 얼마나 빨리 태스크를 완성하는가?
- 사용에 대한 기억(memorability): 시스템에 대한 사용경험이 있는 사용자가 다시 사용할 때 얼마나 기억할 수 있는가? 혹은 다음 번 사용을 위해서 또 다시 학습이 필요한가?
- 실수의 빈도 및 심각성(error frequency and severity): 시스템을 사용하면서 얼마나 자주 실수를 저지르고 있는가? 이러한 실수들은 심각한 것들인가? 또한 그 실수들을 해결하기 위해 사용자들은 어떠한 방법을 택하는가?
- 개인의 만족도(subjective satisfaction): 사용자가 시스템 사용을 얼마나 좋아하는가?

사용성에 대해 가장 표준적이고 포괄적인 정의는 ISO 9241-11[14]에서 제시되어 있는데, 여기서 사용성이란 어떤 상품이 특정한 사용자가 특정한 목적을 달성함에 있어 효과적이고, 효율적이며, 특정한 사용 상황에서 만족할 수 있는 것이라고 정의하고 있다.

Jokela와 Iivari[19]는 사용성을 효율성, 효과성, 만족 등의 속성 집합과 사용 상황으로 구성된 함수로 정의하고, 다음과 같이 나무 형태의 그림으로 사용성을 설명하고 있다.



(그림 1) 사용성의 정의

이 정의는 매우 복잡한 개념으로 제공되기는 하지만, 사용자의 목적을 달성하기 위한 기능과 거기에 포함되는 중요한 속성에 대한 단서를 제공하므로, SW 적응력 개념 정의에 시사점을 제공한다.

### 2.3. SW 적응력(adaptability)

SW 산업의 발전으로 다양한 응용 SW들이 쏟아져 나오고 있기 때문에, 업무를 편리하게 처리하는데 도움이 될 수도 있지만, 새로운 응용 SW를 익혀야 하는 부담은 계속 증가할 수밖에 없다.

따라서 새롭게 개발되거나 업그레이드된 SW 기능을 가르치기보다는, 변화되는 SW 상황에 적응할 수 있는 능력을 길러주는 것이 중요하다. 이러한 관점에서 본 연구자는 앞서 살펴본 SW 사용성 개념에 기초하여 SW 적응력이라는 용어를 제안하고자 한다.

SW 적응력은 문제 상황에서 학습자 스스로 SW의 적절한 기능을 쉽게 찾아 학습하고, 이를 문제 사태에 적용, 효율적으로 해결할 수 있으며, 과지와 전이를 촉진시킬 수 있는 능력이라 정의하고, 이를 배양하는 것을 SW 교수·학습 방법 개선의 목표로 삼는다.

SW 사용성은 개발자의 입장에서 어떻게 설계해야 할 것인가에 관심을 갖는 반면, SW 적응력은 이와는 반대로 사용자의 입장에서 어떻게 학습해야 할 것인가에 대한 개념이라고 할 수 있다. 이는 다음 그림과 같이 도식화 할 수 있다.



(그림 2) SW 사용성과 적응력 개념도

HCI(Human Computer Interaction) 영역에서 주로 개발자 측면에서 사용성에 초점을 맞추어 논의를 진행시켜왔다고 본다면, 컴퓨터교육 측면에서는 반대로 사용성을 고려하여 개발한 SW에 보다 쉽게 적응할 수 있도록 교수·학습 전략을 수립할 필요가 있다는 것이 본 연구의 아이디어이다.

한편, SW 적응력 신장은 SW를 기능 중심으로 분석하고 이를 학습하는 방안과, 해당 SW를 이용한 작업의 유형을 도출하고 이를 중심으로 학습하는 방법이 있을 수 있다. 본 연구에서는 전자를 중심으로 다룬다.

### 3. GUI에 기초한 SW 학습 전략

SW 적응력 배양을 위해서는 UI(User Interface)에 대한 이해가 필수적으로 요구된다. 특히 최근 UI는 대부분 GUI(Graphical User Interface)로 정착되었기 때문에 GUI에 대한 정확한 이해와 활용 능력이 필요하다. 또한, UI 디자인 원칙을 이해하고 이에 따른 SW 학습 전략을 수립하고 실행할 수 있어야 한다.

### 3.1. GUI의 개념과 구성 요소, 그리고 시사점

GUI의 개념은 다양한 정의가 있으며, 이에 따라 그 구성요소도 각기 다르게 정의되기도 한다. 네이버 백과사전[3]은 사용자가 그래픽을 통해 컴퓨터와 정보를 교환하는 작업 환경을 이르는 말, 브리태니커 사전[4]에서는 그림 문자(아이콘) 등을 사용해 직관적인 컴퓨터 조작이 가능하도록 고안된 디스플레이 형식의 인터페이스라고 정의하고 있다.

Webopedia[20]를 보면, GUI는 프로그램 인터페이스로 컴퓨터 그래픽의 역량으로 프로그램을 사용하기 쉽게 만들어주며, 잘 설계된 GUI는 복잡한 명령어를 배우지 않아도 쉽게 사용할 수 있도록 해준다고 정의하고 있다.

이런 종류의 정의는 주로 일반 사용자 입장에서 GUI 용어의 이해를 도모하는 차원에서 이루어진 것으로 보인다. 반면에 다음과 같이 주로 개발자의 입장에서 용어를 정의한 것도 있다.

GUI는 실제 생활과 비슷한 메타포를 사용한 객체(objects; 버튼, 메뉴 등)로 구성되어 있다. GUI는 이벤트의 집합으로부터 사용자와 시스템이 만든 이벤트를 입력받아 결과를 만들어내는 계층적, 그래픽 환경의 SW이다. GUI는 그래픽 형태의 객체(objects)로 구성되어 있으며, 각기 속성(properties)의 집합을 갖고 있고, 속성은 GUI 실행 중에 GUI의 상태를 구성하는 서로 다른 값을 가지고 있다[12].

이런 관점의 차이에 따라 구성 요소의 정의도 다음과 같이 차이를 보이기도 한다.

#### 사용자 입장의 GUI의 구성 요소 정의 예[22]

- 포인터(pointer): 스크린에서 객체를 선택하고 명령을 실행시키기 위한 심볼
- 입력장치(pointing device): 마우스나 트랙볼 같은 장치
- 아이콘(icons): 명령, 파일, 창 등을 나타내는 작은 그림
- 데스크탑(desktop): 실제 객체들을 가리키는 아이콘 등이 위치하는 스크린
- 창(windows): 스크린을 서로 다른 영역으로 나누어 작업할 수 있는 영역
- 메뉴(menus): 선택해서 실행할 수 있도록 해주는 명령

#### 개발자 입장의 GUI의 구성 요소 정의 예[15]

- 컨트롤(controls or components or widgets): 버튼, 텍스트 박스 등

- 창(a windows or form or frame): 서로 다른 컨트롤을 포함하고 있는 스크린 영역
- 메뉴(a menu bar): 컨트롤과 비슷하나 다른 방법으로 기능을 제공하는 막대
- 형태(layout): 컨트롤의 위치 등을 규정
- 상호작용(interaction): 마우스 등을 이용하여 컨트롤과의 상호작용

이상과 같은 GUI의 개념 정의와 구성 요소 고찰에서 다음과 같은 시사점을 얻을 수 있다.

첫째, GUI의 환경과 실생활에 대한 매핑 훈련을 통한 일반화의 필요성이다. GUI 개념 정의에서 핵심은 실생활과 비슷한 그래픽 환경의 제공 및 상호작용이다. 직관적 조작이 가능하도록 실생활을 모방한 것이다. 따라서 GUI 환경에서 실생활과 유사한 조작으로 SW 기능 실행이 가능한 것을 보여주고, 학습자에게 이러한 활동을 격려하고 조장할 필요가 있다. 예를 들면, 물건을 해당 용기에 넣는다라는 실생활의 조작은 다음과 같이 매핑되고 일반화시킬 수 있다.

- 매핑 예: 탐색기의 파일을 휴지통에 드래그 앤드 드롭하여 넣는다(지운다).
- 매핑 예: 한글 데이터 파일을 한글 바로가기 아이콘이나, 열려 있는 한글 프로그램에 드래그 앤드 드롭하여 넣는다(파일을 열어준다).
- 일반화: 어떤 파일을 해당 SW에 넣으면(마우스 드래그 앤드 드롭), 그 파일에 대한 작업을 수행할 수 있다.

둘째, GUI 컨트롤(objects, controls, components, widgets)에 대한 정확한 이해와 조작 방법에 대한 훈련의 필요성이다. GUI의 개념과 구성 요소를 살펴보면 결국 컨트롤과 사용자가 어떻게 상호작용 하는가 하는 문제로 귀착되고, 사용자 입장에서는 컨트롤을 정확하게 이해하고 조작하는 능력이 필요하다.

### 3.2. UI 디자인을 위한 원칙

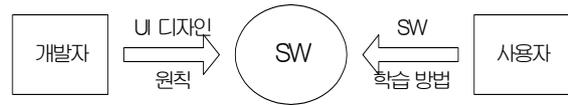
HCI 특히, 사용자 인터페이스 측면에서 최근 사용되는 SW 디자인의 방향이나 원칙에 대한 고찰은 SW 학습에 시사점을 줄 수 있다.

UI 설계 분야에서 명성이 높은 애플 컴퓨터社[11]에서 제시한 UI 디자인을 위한 원칙(The Human Interface Design Principles)에서 SW 학습과 관련된 있는 요소를 추출하여 정리하면 다음과 같다.

- 메타포(metaphor): 구체적이고 친숙한 개념을 표현하는 메타포를 사용하여 사용자가 컴퓨터 환경에 대한 예상할 수 있도록 한다. 예를 들어, 문서를 저장하기 위해 파일 폴더의 메타포를 사용한다. 즉, 파일 캐비닛을 정리하는 방식과 유사하게 하드디스크를 정리할 수 있다. 실생활의 대상물들을 컴퓨터 화면으로 옮긴 아이콘들은 실세계의 메타포로서 인터페이스에 효과적으로 작용하며 문화적, 언어적 장애들을 말보다 더 잘 극복할 수 있게 해준다.
- 직접조작(direct manipulation): 직접 조작은 사용자로 하여금 컴퓨터가 제공하는 객체를 직접 관리할 수 있도록 한다. 이러한 원리에 따라, 스크린상의 객체는 사용자가 객체에 대한 동작을 수행하는 동안 시각적으로 보여야 하며, 그 동작의 결과는 즉시 나타나야 한다.
- 보고 선택하기(see and point): 사용자는 장치를 사용하여 스크린상의 객체를 선택함으로써 인터페이스와 상호 작용한다. 사용자가 자신이 화면 상에서 작업하는 내용을 항상 볼 수 있고, 자신이 보는 것을 선택할 수 있어야 한다.
- 사용자 컨트롤(user control): 컴퓨터가 아닌 사용자가 동작을 초기화하고, 제어할 수 있도록 한다. 사용자에게 작업을 완성할 수 있도록 하고, 최소 불가능한 동작을 사용하지 않도록 도와주어야 한다.
- 반응과 대화(feedback and dialog): 제품을 사용하면 어떤 현상이 발생하는 지 사용자에게 알려준다. 즉, 사용자가 작업을 수행할 때 피드백을 제공한다.
- 관대함(forgiveness): 대부분의 동작을 간단하게 취소할 수 있도록 애플리케이션을 개발해야 한다. 사용자는 시스템에 손상을 주지 않고, 작업을 시도할 수 있어야 한다. 예를 들면 취소 및 복귀 커맨드와 같은 안전한 망을 만들어 배우고 사용하는 것이 편리하다고 느낄 수 있어야 한다.

### 3.3. UI 디자인 원칙에 따른 SW 학습 전략

SW 개발자는 UI 디자인 원칙에 따라 SW 설계하고 개발한다. 사용자는 SW 내부에는 접근할 수 없으나 UI와는 직접 부딪히면서 사용법을 배운다. 이때 원칙 없는 시행착오식 접근보다는, UI 디자인의 원칙을 이해하고 어떤 절차나 방법에 따라 접근한다면 보다 효율적인 학습이 가능할 것이다. 즉, 다음 그림과 같이 사용자와 개발자는 서로 반대의 입장에 서 있으나 결국 SW 측면에서 본다면 거의 유사한 관점으로 SW에 접근할 수 있을 것이다.



(그림 3) SW UI 디자인 원칙과 SW의 학습 방법

앞서 살펴본 UI 디자인 원칙으로부터 도출할 수 있는 SW 학습 전략은 다음과 같다.

#### 3.3.1. 기능 실행에 대한 암시 파악

메타포는 아이콘 등을 통해 표현되기도 하지만, 마우스 포인터를 통해서도 제시된다. 마우스 포인터의 모양은 어떤 작업(조작)이 가능한지를 암시해 준다. 따라서, SW를 사용할 때는 마우스 포인터의 위치와 이동, 조작에 따라 포인터의 모양이 어떻게 변하는지 정확히 알고 있으면 조작을 보다 쉽게 할 수 있다. 또한 반대로 마우스 포인터나 커서의 위치에 따라 수행할 수 있는 기능이 달라진다. 따라서 커서나 마우스 포인터로 작업 대상을 정확히 선택하고, 이에 따른 변화를 주의 깊게 관찰한다.

#### 3.3.2. 실생활의 모델링

직접조작(direct manipulation)과 메타포의 결합 예는, 가령 휴지통에 휴지를 버리듯이 삭제할 대상을 마우스로 끌어서 휴지통 아이콘에 올려놓는 경우 해당 파일이 삭제되는 것이다. 또는 탐색기 파일 목록의 출력 순서를 바꾸기 위해서 목록단추를 끌어서 위치를 이동시키는 것 등의 예를 들 수 있다. 이러한 원리는 사용자가 SW의 사용법을 몰라도 실생활과 유사한 조작을 통해서 기능 수행이 가능하다는 것을 의미한다. 따라서 이와 같이 실생활을 의도적으로 모델링하여 SW 기능 수행을 시도해 본다.

#### 3.3.3. 단축메뉴의 활용

보고 선택하기(see and point)는 컴퓨터와 사람간의 가장 일반적인 상호작용이다. 그러나 일반적인 만큼 사용자들에게는 매우 추상적일 수도 있다. 보고 선택하기로 설계된 SW에서 가장 효율적인 접근 전략으로 단축메뉴의 호출을 제시할 수 있다. 작업의 대상에서 마우스의 오른쪽 버튼을 누르게 되면 대부분의 SW에서는 실행 가능한 메뉴의 목록이 나타나고, 여기에서 필요한 대부분의 작업을 실행할 수 있다.

#### 3.3.4. 직관에 따른 시도

사용자 컨트롤(user control)의 원칙이 잘 지켜진다면 사용자의 조작 미숙에 따라 치명적인 결과를

초래하지 않을 것이다. 또한 관대함(Forgiveness) 원칙은 하나의 기능을 수행하는 것에 있어 여러 가지 방법을 제공하며, 아주 정교한 조작을 하지 않아도 기능이 수행될 수 있도록 SW가 설계됨을 의미할 수 있다. 따라서 사용자는 SW 사용법을 몰라도 자신감을 가지고 추정된 방법대로 시도할 경우 성공 확률이 높을 수 있다.

이때 반응과 대화(Feedback and Dialog) 원칙에 따라, 조작에 따른 응답이 주어지므로 주의 깊은 관찰이 동반되는 경우에는 성공 확률이 더 증가할 것이다.

### 3.3.5. 효율성 추구

SW의 활용 능력의 제고는 결과의 도출뿐만이 아니라 얼마나 효율적으로 작업을 수행할 수 있는가가 중요하다. 똑 같은 작업 결과를 얻더라도 최소한의 조작으로 결과를 얻을 수 있는 효율성 측면을 고려해야 한다. 따라서 과업의 완수 후에는 반성의 단계를 거쳐 보다 효율적인 수행 방법은 없었는지 검토하는 과정이 필요하다.

## 3.4. UI 디자인 원칙에 따른 SW 학습 단계

이상에서 고찰된 UI 디자인 원칙에 따른 SW 학습 전략을 바탕으로 다음과 같이 일반화된 SW 학습 단계를 도출할 수 있다.

- 1 단계: 작업 대상 선택
- 2 단계: 직관적 조작
- 3 단계: 단축메뉴 호출

이는 GUI SW 설계 원리에 따라 SW 학습이 이루어져야 효율적이라는 관점으로, 학습자 스스로 기능 수행 방법을 찾아내고 SW 적응력을 신장시키기 위한 문제해결 접근 단계이다. 이는 단순하지만 GUI SW에 대한 접근 방식을 정형화시킴으로써 학습자들이 보다 손쉽게 SW 기능 사용법을 탐구할 수 있도록 도와줄 수 있을 것이다.

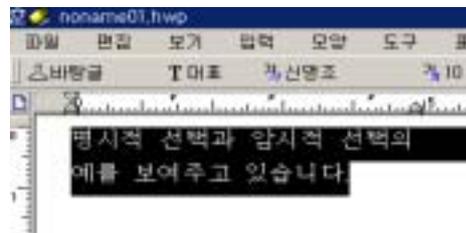
### 3.4.1. 1단계: 작업 대상 선택

기능 실행에 대한 암시 파악 전략으로부터 이 단계를 유도할 수 있다. 어떤 기능을 실행하기에 앞서 작업 대상을 선택해야만 한다. 대상을 선택해야만 해당 기능에 대한 암시를 받을 수 있다. 또한, 특정한 메뉴나 도구상자의 기능은 대상을 선택하지 않으면 사용할 수 없는 경우도 있다.

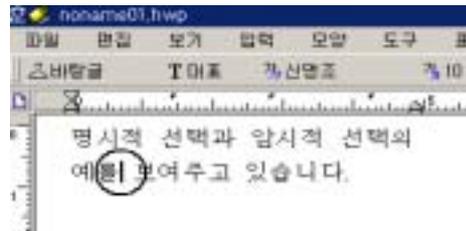
한편, 대상을 선택하는 방법은 명시적 선택과 암시

적 선택이 있다. 명시적 선택은 마우스 클릭하거나 드래그(블록설정)하여 작업 대상을 선택한다. 선택된 대상은 반전되거나 파란색으로 변하거나, 또는 선택 영역이 표시된다. 암시적 선택은 커서나 마우스 포인터를 작업 대상의 위에 올려놓는 것으로 작동한다.

한글에서 문단 모양을 지정하는 작업에서 명시적 선택의 예를 들면, (그림 4)와 같이 문단 전체를 블록으로 설정한 후, '문단 모양'을 지정하는 방법이다. 이에 반해 암시적 선택은 (그림 5)와 같이 커서를 해당 문단에 가져다 놓은 후에, '문단 모양'을 지정하는 방법이다.



(그림 4) 명시적 선택(블록 설정)



(그림 5) 암시적 선택(커서의 위치)

사용자들은 종종 작업대상을 선택하지 않음으로 인해 SW 기능 실행에 장애를 겪곤 한다. 아동의 ICT 활용 오류 유형 연구[2]를 보면, 활성화창과 비활성 창의 의미를 혼동하는 오류가 많음을 알 수 있는데, 이는 아주 기본적인 작업 대상의 선택에 있어서도 장애가 있음을 보여주는 예라고 할 수 있다. 특히 암시적 선택은 사용자들이 별로 의식하지 않은 상태에서 이루어지게 되므로 이에 관련된 오류가 더 많을 것으로 추정된다.

따라서, 거의 모든 작업이 대상을 선택하는 것으로부터 출발하므로, SW 기능을 실행하기에 앞서 대상을 선택하는 것을 첫 번째 단계로서 접근할 필요가 있다.

### 3.4.2. 2단계: 직관적 조작

기능 실행에 대한 암시 파악, 실생활의 모델링, 직관에 따른 시도 전략으로부터 이 단계를 유도할 수

있다. 1단계에서 작업 대상을 선택하면, 어떤 암시를 포착할 수 있으며, 이를 바탕으로 실생활을 그대로 모델링하여 SW 기능 실행에 적용할 수 있다. 직접 조작 원리에 따라 사용자가 컴퓨터에 의해 묘사된 객체를 직접 제어하는 느낌이 들도록 SW를 설계하며, 여기에 메타포의 원칙이 결합되어 더욱 쉽게 그 기능을 실행할 수 있기 때문이다.

즉, 사용자의 조작 방법을 미리 예측하여 쉽게 사용할 수 있도록 설계되어 있으므로, 바로 이 점을 이용하여 ‘이렇게 하면 될 것 같다’라고 직관적으로 느끼는 대로 실행해본다. 이 때 구체적인 실행 방법은 작업 대상을 상대로 클릭, 더블클릭, 드래그, 드래그 앤 드롭과 같은 마우스 조작이다.

따라서 1단계에서 작업 대상을 선택한 후, 2단계에서는 이와 같은 몇 가지 마우스 조작을 실행해 본다.

### 3.4.3. 3단계: 단축메뉴 호출

GUI에서 시스템과 상호작용하는 가장 보편적인 방법이 보고 선택하기(see and point) 원리에 따른 ‘단축메뉴’를 활용이다. 그러나 의외로 많은 사용자들이 ‘단축메뉴’의 호출에 익숙하지 않은 경향이 있다.

작업 대상을 선택한 후 마우스 오른쪽 버튼을 누르면 ‘단축메뉴’가 호출된다. 여기에서는 작업 대상에 따라서 각기 다른 기능이 제시된다. 예를 들어 바탕화면, 작업표시줄, 단축아이콘에서 오른쪽 버튼을 누르면 각기 다른 ‘단축메뉴’가 호출된다. 이는 현재 선택된 대상에서 할 수 있는 기능이 서로 다르기 때문이다.

따라서 1단계에서 작업 대상을 선택한 후, 2단계에서 처리를 하지 못하면, 3단계로 진행하여, ‘단축메뉴’를 호출하고 자신이 처리하고자 하는 기능이 있는지 살펴보고 실행시키는 것은 매우 중요한 단계이자 효과적인 전략이다.

## 4. SW 적응력 향상을 위한 탐구적 교수·학습

### 4.1. 시범·실습식 수업과 SW 교수·학습

시범(demonstration)은 학습해야 할 기능이나 절차의 실제적인 사례를 제시하고, 학습자가 이를 관찰하는 것이며, 실습은 시범을 통하여 익힌 것을 교수자의 통제 하에 직접 연습하고 적용해 보는 방법이다[6].

시범과 실습은 사회적 인지이론의 관찰학습 방법, 행동주의적 학습 이론에서 반복 연습과 피드백 방법에서 그 이론적 토대를 형성할 수 있다. 관찰학습은 사회적 인지이론에서 모델링의 역할과 연계되어 논의되는 방법으로, 학습자가 모델링을 통하여 얻어진

지식을 이용하여 새로운 행동의 변화를 일으켰을 때 일어난다. 관찰학습 과정은 학습자에게 유의미한 행위에 대한 주의 집중(attention), 습득한 지식을 내적, 외적으로 시연하고 재현하여 개념 모델을 형성하는 과제(retention), 형성된 개념 모델에 따라 행동을 수행하는 생산(production), 행위의 기능적 가치와 적절성을 인지함으로써 동기화(motivation) 과정을 거친다[10].

과정적 접근에서는 학생들에게 실행을 연습시키기 전에 앞서 학습될 상이한 과정 또는 기술을 교사가 시범을 보이는데[9], SW 학습에서는 대부분 이 방법을 사용하는 것으로 추정된다.

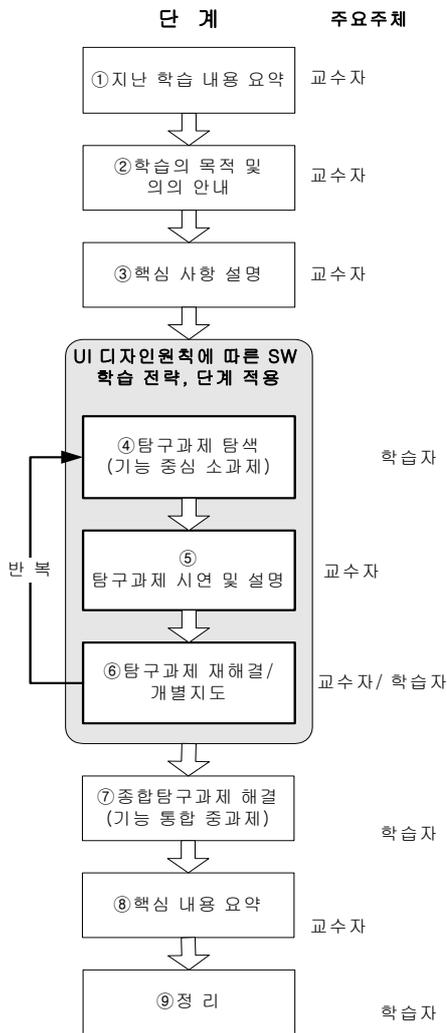
프레젠테이션 장비의 고성능화와 더불어 가격의 하락은 이러한 경향을 더욱 부추기게 된 결과를 초래했다고 추정할 수 있다. GUI 방식의 SW 사용법을 가르치고 배우기 위해서는 Visual 한 방법이 동원되는 것은 지극히 당연하고도 꼭 필요한 요소이기도 하지만 반성해야 할 점도 있다. 가령, 수학 문제를 해결함에 있어서 해답을 먼저 보고 푸는 것과, 해답 없이 문제에 먼저 부딪치고 고민하다 나중에 해답을 보는 것은 매우 다른 학습 결과를 가져올 수 있다. 시범·실습식 SW 교수·학습 방법의 가장 큰 문제점은 바로 이와 같은 점에서 찾을 수 있다.

최동근 외[9]는 시범·실습식 수업의 단점으로 학생들에게 그들 스스로 어떤 것을 발견하고 문제를 해결하게 하는 기회가 적게 제공되고, 학생들은 주로 관찰자가 되므로 능동적 참여의 기회가 줄어들며, 학급에서 듣고 보는 데 치중하는 등의 문제가 발생할 수 있다는 등을 들고 있는데, 이는 기존의 시범·실습식 SW 교수·학습에 대해 반성할 필요성을 제기한다.

### 4.2. SW 적응력 향상을 위한 탐구적 교수·학습 모형

기존의 시범·실습식 SW 교수·학습 방법에 대한 대안으로써, 학습자 스스로 탐구하는 학습 과정을 통해 SW 적응력을 향상시킬 수 있는 모형을 다음과 같이 제시한다.

#### 4.2.1. 모형의 단계



(그림 6) SW 적응력 향상을 위한 탐구적 교수·학습 모형

#### 4.2.2. 단계별 주요 활동

- ① 지난 학습 내용 요약
- ② 학습의 목적 및 의의 안내  
학습 목적과 내용의 의미를 학습자가 충분히 이해하도록 하여 학습 의욕을 고취시킨다.
- ③ 핵심 사항 대한 설명  
필수적인 지식만을 제공하고 학습자에게 탐구의 기회를 제공한다.
- ④ 탐구과제 탐색(기능 중심 소과제)  
앞 절에서 제시한 UI 디자인 원칙에 따른 SW 학습 전략과 단계가 적용된다. SW 기능 분석에 따라 수준별, 문제해결중심의 소과제가 제시된다. 교수자의 설명 없이 학습자 스스로 탐구

하며, 이때는 질문을 받지 않는다. 스스로 문제를 해결하려는 시도가 자체를 중요하게 여긴다. 이 과정에서 학습자가 화면구성, 메뉴 등에 익숙해지게 하는 부수적 효과도 있다.

- ⑤ 탐구과제 시연 및 설명
- ⑥ 탐구과제를 다시 해결 및 개별지도  
교수자의 설명을 듣고 다시 과제를 실습해보며, 이때 교수자는 질문을 받고 개별 지도를 한다. 탐구과제가 여러 개이므로 ④번 과정부터 다시 반복하여 수행한다.
- ⑦ 종합 탐구과제 해결  
④의 기능 중심 소과제에서 학습한 기능을 통합하여 종과제 정도의 문제를 해결한다.
- ⑧ 핵심 내용 요약
- ⑨ 정리

마지막 단계에서는 과제를 통하여 얻어질 수 있는 일반적인 원리를 찾아내어 정리하거나, 과제 수행에 대한 반성을 한다. 또한 이 단계에서 연습의 기회를 충분히 제공하고, 다른 상황으로의 전이를 유도할 수 있도록 한다.

#### 4.2.3. 모형의 특징과 해설

본 모형의 가장 큰 특징은 첫째, 앞 절에서 제시한 UI 디자인 원칙에 따른 SW 학습 전략과 단계를 적용한다는 것이다. 이는 학습자 중심의 탐구 기회를 제공하고 궁극적으로 학습자의 SW 적응력을 높이기 위한 목적이다.

이는 문제해결학습의 특징과도 맥을 같이 하는데, 문제해결학습의 특징을 다음과 같이 요약할 수 있다[1].

첫째, 관련분야에 실재하는 복잡하고 비구조적인 문제들을 풀어 가는 것이다. 그러나 문제해결안을 도출해 내는 것도 중요하지만 그 결과 자체만 보는 것이 아니고 해결해 가는 과정 자체에도 주안점을 두고 있다. 이 과정을 통해서 학습자들이 문제해결학습에서 학습목표로 내세우고 있는 관련분야의 문제해결 및 비판적 사고력을 습득할 수 있다고 보기 때문이다.

둘째, 학습자 중심의 수업형태이다. 학습자 자신의 학습이 얼마나 이루어졌는지 그리고 무엇이 앞으로 더욱 필요한지를 스스로도 인식하고 깨달을 수 있도록 하자는 것인데 이는 단지 학습의 효과 면에서만 뿐만 아니라 스스로에게 비강압적인 방법에 의한 학습의 내적 동기부여라는 측면에서도 중요하며 나아가 궁극적으로 모든 기업이 추구하고자 하는 체질화된 학습인으로서의 태도를 습득하게 하는데도 영향을 미칠 수도 있다.

그런데 기존의 시범·실습식 수업과 같이 ‘무작정 따라하기 식’의 수업은 문제해결학습과는 상당히 거리가 먼 것으로 보인다.

이와 같은 문제의식과 문제해결학습의 관점에서 이 모형은 ④, ⑤, ⑥ 단계를 통해 학습자가 먼저, 그리고 스스로 탐색하는 과정을 강조한다.

SW 기능에 대한 교수자의 우선적 설명과 그에 따른 실습을 지양하고, 교수자의 설명 없이 학습자의 탐구를 먼저 유도함으로써 학습자 스스로 SW 적응력 신장을 배양할 수 있도록 유도한다. 결과가 아니라 과정 자체를 중시하는 것이다.

한편 이 부분에서 추가로 고려된 사항이 있다. 교수자는 매우 익숙한 SW, UI이기 때문에 능숙한 시범을 보일 수 있지만, 학습자는 낯선 것일 수 있으므로 바로 시연에 들어가게 되면 어리둥절해지기 쉽다는 문제가 있다. 그런데 이 모형에서는 학습자가 시범이나 설명을 듣기에 앞서, 자기의 학습 속도에 맞게 SW의 기능을 탐색하는 과정 속에서 SW 환경에 익숙해지도록 해줌으로써, 시연시 SW에 대한 친근감을 주고 아울러 처리 절차와 방법에 대한 인지적 부담을 덜어 줄 수 있는 장점도 있다.

두 번째는 응용 SW 기능 분석을 통한 수준별 탐구과제 제공이다. ④ 단계에서 실습과제를 제시할 때, SW의 기능을 수준별로, 단순 기능에서 고급 기능으로 점차적으로 제공하는 것이다. 또한 하나의 기능에서 여러 가지 기능을 통합하여 해결할 수 있는 과제를 제시함으로써 기초기능으로부터 출발하여 점차 기술의 완성도를 높일 수 있도록 한다. 이와 같은 전략은 학생들이 좌절감을 적게 느끼고 도전감을 줄 수 있도록 도와줄 것이다.

세 번째는 문제해결 중심의 기능 과제의 제시이다. SW의 메뉴별 기능이 아닌, 해당 SW를 활용한 문제 해결을 위한 기초 기능을 분석하여 제공하는 전략이다. 메뉴별로 제공되는 기능을 차례대로 익히는 방법은 학습자가 각 기능이 갖는 의미나 활용 방안을 충분히 파악하지 못함으로써, 정작 문제를 해결하는 과정에서 적절히 그 기능을 활용하지 못할 수 있다. 따라서 문제상황에서 이를 해결하는데 필요한 기능을 추출하고, 이를 학습하는 전략이 보다 바람직할 것이다. 그리고 ⑤ 단계에서 점차로 문제해결을 위해 기능을 통합하여 활용하는 능력 배양을 추구한다.

네 번째는 반복과 연습을 통한 자동화이다. 이는 ④, ⑤, ⑥ 단계와 ⑨ 단계를 통해 이루어진다. 반복 연습 전략은 새로운 기능을 숙지하기 위한 효과적인 과정이 되며, 이 과정을 통하여 학습자는 기능의 절차적 수행에 대한 지적, 양적인 증진을 얻을 뿐만 아

니라, 이에 관련된 지적인 기능도 확고히 확립할 수 있다. 한편, ④, ⑤, ⑥ 단계를 통해 학습자의 수행에 대한 점진적인 피드백 제공 잘못 형성된 개념과 기능을 수정하여 정확한 수행을 가능하게 한다.

## 5. 결론

SW 교수·학습에 있어 대부분의 교사들이 단지 교재에 의존하거나 프로젝트, 프로젝트 TV 등을 이용한 단순한 시범과 실습 등에 의존하고 있는 형편인데, SW는 끊임없이 새롭게 개발, 업그레이드되기 때문에 이와 같은 방법의 파지와 전이 효과는 매우 의심스럽지 않을 수 없다.

이와 같은 문제의식을 가지고 학습자 스스로 문제 상황에서 SW의 적절한 기능을 쉽게 찾아 학습하고, 이를 문제 사태에 적용, 효율적으로 해결할 수 있으며, 파지와 전이를 촉진시킬 수 있는 능력을 SW 적응력이라 정의하고, 이를 배양할 수 있는 SW 교수·학습 방법을 탐색하였다.

지식정보사회에서 요구하는 인간상은 창의적 문제 해결력을 갖춘 지식전문가이다. 이런 인간을 양성하기 위해서는 SW 학습 방법에 있어서도 창의성과 문제해결력을 강조해야 할 것이다. 본 연구에서 제시한 전략은 UI의 설계 원리를 바탕으로 학습자의 탐구력을 강조하며, 자기주도적이고 효율적인 SW 학습을 추구한다. GUI기반의 응용 SW는 사용자의 직관적 조작에 따라 그 기능을 수행할 수 있도록 설계되어 있기 때문에, 학습자 스스로 SW의 기능 실행 방법을 찾아내고 실행하는 것이 가능하며, 이렇게 직관적으로 SW에 접근하는 방법이 새로운 SW에 대한 적응력을 높여줄 수 있다는 관점이다. 또한 세 단계로 이루어진 SW 학습 단계는 학습자가 수행할 수 있는 구체적 작업의 일반화를 제공함으로써, 보다 수월하게 SW 기능 수행 방법을 찾아낼 수 있도록 도와줄 수 있을 것이다.

한편, 본 연구에서 제안한 SW 적응력 향상을 위한 탐구적 교수·학습 모형은 학습자 스스로 SW 적응력을 배양할 수 있도록 유도하고 문제해결력 신장을 강조하고 있다. 기존의 SW 학습 패러다임은 여러 가지 문제에 직면하고 있는데, 이에 본 연구는 SW 학습의 새로운 방향을 모색하는 하나의 단초를 제공할 것이다.

향후 연구 과제로는 본 연구에서 제안한 SW 적응력 향상을 위한 교수·학습 전략을 적용한 현장 사례 연구를 들 수 있다.

## 참고문헌

- [1] 강인애(1997). 왜 구성주의인가?. 문음사.
- [2] 김은옥(2003). 아동의 ICT 활용 오류 유형 분석 및 처치 방안. 교육학석사학위논문. 대구교육대학교 교육대학원.
- [3] 네이버(2004). GUI. 네이버 백과사전.  
<http://100.naver.com/100.php?where=100&id=717584>
- [4] 브리태니커 온라인(2004). GUI. 한국 브리태니커 온라인.  
[http://preview.britannica.co.kr/bol/topic.asp?article\\_id=rts02g019](http://preview.britannica.co.kr/bol/topic.asp?article_id=rts02g019)
- [5] 손상희(2000). 웹에서의 인터페이스 디자인에 관한 고찰. 2000 한국디자인 포럼, Vol.5 No.1, pp.170-197.
- [6] 이무근, 김재식, 김관욱(2000). 실기교육방법론.
- [7] 이성흠, 권숙진, 이영민(2000). 효과적인 컴퓨터 기반 학습 프로그램 개발을 위한 편의성 감사의 이론적 기초. 교육공학연구 제16권 제1호. pp.163-179.
- [8] 이태욱, 유인환, 이철현(2001). ICT 교육론. 형설.
- [9] 최동근, 양용철, 박인우(2001). 교육방법의 공학적 접근. 교육과학사.
- [10] 허희옥 외(2001). 컴퓨터교육방법 탐구. 교육과학사.
- [11] Apple Computer, Inc(1996). The Human Interface Design Principles.  
<http://developer.apple.com/documentation/mac/HIGuidelines/HIGuidelines-16.html>
- [12] Atif M. Memon, Mary Lou Soffa, Martha E. Pollack(2001). Coverage criteria for GUI testing. Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering, Volume 26 Issue 5, pp.256-267.
- [13] Franca Garzotto, Maristella Matera, Paolo Paolini(1998). Model-based heuristic evaluation of hypermedia usability. Proceedings of the working conference on Advanced visual interfaces, pp.135-145.
- [14] ISO/IEC(1998). 9241-14 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 14 Menu dialogues, ISO/IEC 9241-14: 1998 (E).
- [15] Judith Bishop, Nigel Horspool(2004). Developing principles of GUI programming using views. Proceedings of the 35th SIGCSE technical symposium on Computer science education, pp.373-377.
- [16] Lerone Olfman, Conrad Shayo(1997). The Role of Training in Preparing End Users to Learn New but Similar Software Packages. SG CPR97, San Francisco CA USA, pp. 210-223.
- [17] Susan Wiedenbeck, Patti L. Zila(1997). ACM Transactions on Computer-Human Interaction, Vol. 4, No.2, pp.169-196.
- [18] TeamInterface(2002). usability.  
<http://uidesign.co.kr/academy/ui3sr041.asp>
- [19] Timo Jokela, Netta Iivari, Juha Matero, Minna Karukka(2003). The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. Proceedings of the Latin American conference on Human-computer interaction, pp.53-60.
- [20] webopedia(2004). graphical user interface.  
[http://www.webopedia.com/TERM/G/Graphical\\_User\\_Interface\\_GUI.html](http://www.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html)

## 저자소개

### 유인환



2000 한국교원대학교  
컴퓨터교육과(교육학박사)  
현재 대구교육대학교 조교수  
E-Mail: bluenull@dnu.ac.kr

### 구덕희



2000 한국교원대학교  
컴퓨터교육과(교육학박사)  
현재 대구교육대학교 전임강사  
E-Mail: koo@dnu.ac.kr