

# 템플릿을 사용한 객관식 출제시스템의 설계 및 구현

김진희\* · 용환승\*\*

## 요 약

컴퓨터를 교육적으로 활용하기 위한 다양한 노력들이 시도되고 있으며 이러한 노력의 결과로 각종 학습 프로그램들과 테스트 프로그램들이 제작되고 있다. 이제까지의 테스트 프로그램들은 학습자에게 일방적으로 문제를 내고 풀도록 하는데 그쳤다. 또한, 출제자 역시 학습문제를 일일이 입력해야 했다.

따라서 본 논문은 첫째, 학습자 스스로가 문제를 입력할 수 있도록 한다. 둘째, 이렇게 만들어진 문제를 기초 데이터로 하여 새로운 문제를 만들어내는 시스템을 개발한다. 문제자동출제 알고리즘을 적용시킨 시스템의 구현환경을 소개하고, 각종 내부 모듈과 문제은행, 문제자동출제의 유형별 예제를 들어 설명한다. 데이터를 활용하여 문제를 자동으로 출제해주는 시스템을 개발한다. 이것을 교육용 프로그램에 적용시키는 방안에 대하여 알아보고, 이렇게 설계 구현한 시스템을 한자과목에 도입하여 이의 효용성을 타진해보고 이를 다른 과목에 적용시킬 수 있는가에 대하여 알아본다.

## Design and Implementation of Test Item Generation System based on Template

Jin-Hee Kim, Hwan-Seung Yong

### ABSTRACT

There have been many attempts to educationally utilize a computer. As a result, a number of learning programs and testing programs have been developed. Testing programs developed so far are rather focusing on giving learners questions without interactivity. Question givers also have to feed question data into a computer.

Therefore, this thesis, firstly, explains how learners themselves feed question data into a computer. Secondly, this thesis explains how to develop such a system that can produce new questions based on the question data fed by learners. This paper also introduces the system environment, which features the test item generation system, and explains the system environment by exemplifying internal modules, a test item bank. Besides, we also describe how to apply this system to educational programs and identify whether it is possible to apply it to a Chinese subject and others.

### 1 서 론

일반적인 교육용 소프트웨어들은 출제자가 학습자에게 문제를 일방적으로 주는 선형구조의 소프트웨어

들이다. 학습이 끝나면 그것으로 이 교육용 소프트웨어는 생명을 다하게 된다. 학습자는 새로운 교육용 소프트웨어를 찾아 학습하여야한다. 새로운 소프트웨어를 찾았다고 가정하더라도 교육용 소프트웨어의 특성상 이를 활용하는 방법을 다시 익히려면 많은 시간이 소요된다. 막상 전달하고자 하는 교육적인 효과보다는 이 교육적인 효과를 얻기 위하여 행하여야 하는 일련의 과정을 배우는

\* 정희원 : 명지초등학교 컴퓨터 전임 교사

\*\* 정희원 : 이화여자대학교 컴퓨터학과 교수

논문접수 : 2002년 3월 20일, 심사완료 : 2002년 4월 18일

동안 학습자는 지쳐버리고, 시간적, 경제적인 손실을 감수할 수밖에 없다. 따라서, 학습자 스스로가 문제를 입력할 수 있고, 입력된 문제 데이터를 기초로 하여 새로운 문제를 만들어내는 프로그램을 개발하고자 한다.

본 논문은 기존의 교육용 프로그램에서 교수자 또는 출제자가 학습자에게 일방적으로 지식 정보나 문제를 전달하였다는 데 착안한다. 더불어, 출제되는 문제들이 출제자에 의해 일일이 입력되어야 했다는 데 착안한다. 그래서 학습자 스스로가 문제를 출제하고 이를 기초 데이터로 하여 새로운 문제를 만들어 내는 학습프로그램을 구현하고자 한다. 이로써, 자신이 공부하고 입력한 학습문제를 여러 문제 유형으로 반복하여 학습할 수 있도록 하는 것이 목적이다. 이것은 기존 학습 프로그램의 지속적인 학습내용 업데이트 부재문제를 해결할 것이다. 또한 한 학습자가 입력한 문제를 타 학습자에게 전달할 수 있도록 하여 다양한 문제를 접할 수 있도록 한다. 이로써, 풍부한 학습문제를 제공하고 학습자와 학습자간의 커뮤니케이션을 통해 더욱 높은 학습효과를 얻을 수 있는 기반을 만드는 것이 목적이다. 학습의 내용은 각종 언어 영역 중에서 현대인에게 가장 부족하다고 생각되는 한자(漢字)를 선택한다. 이로써, 고리타분하게 느껴질 수 있는 한자학습에 있어서 흥미도 부재에 의한 교육효과 저하문제를 해결하고자 한다[11, 12].

### 1.1 주요 용어 정의

**문제 유형** - 프로그램의 처리를 위해 분류한 문제들의 타입(type)을 말한다. 문제의 변수에 따라서 결정된다.

**문제** - 최종적으로 화면에 출력되는 문제를 말하는 것이 아니라, 프로그램 상에서 처리할 일정한 형식을 갖추고 있는 데이터를 의미한다. 이것에 문제 데이터가 대입되면 최종 문제가 만들어지게 된다.

**문제 데이터** - 변수에 대입될 각종 데이터를 의미한다. 예시, 보기, 답 등에 활용된다.

### 1.2 기존의 자동 문제 출제 시스템 비교

자동 문제 출제에 관련한 기존 연구로는 [13, 14]이 있다. '웹 기반 자동 문제 출제 시스템'[13]에

서는 간편한 문제 입력으로, 난이도에 따른 단순한 알고리즘을 통해 문제를 출제 해 줄 수 있게 하는 시스템을 연구하였다. 그러나, 입력한 문제 외에는 다른 문제를 만들어 내지 못하는 시스템이므로 본 논문에서의 템플릿을 활용한 문제의 자동 생성과 문제의 출제를 함께 하는 시스템과는 비교할 수 있는 부분이 없었다.

## 2. 문제자동출제 시스템의 설계 및 구현

### 2.1 문제자동출제 알고리즘의 설계

현재 출제되고 있는 문제의 유형을 분석한 결과 "예시"부분과 "문제"부분으로 구성되어 있는 것이 많다는 것을 알게 되었다. 문제 유형은 "문제 데이터"와 "문제", 그리고, "보기"와 "답"으로 이루어져 있다 라고 가정한다. 이러한 문제들의 최소단위로 나누어 대치 할 수 있는 부분을 대치한 후 이를 다시 조합하는 알고리즘을 도입해보도록 한다. "문제 데이터"는 "예시" 부분을 말하고 다음을 예로 들어 설명해 보자.

ex) 'ㄱ'의 발음은?

위와 같은 문제의 유형은 예시부분 'ㄱ'와 문제부분 '의 발음은?'으로 구성된다. 이를 변수에 대입하면 다음과 같이 표현할 수 있다.

ex) \$A = "ㄱ"

\$B = "의 발음은?"

여기서 \$A에 다른 예시를, \$B에 다른 문제를 자동으로 넣고 출제 해 주는 알고리즘을 설계하였다. 즉, \$A를 대치하여 "'ㄷ'의 발음은?"을 출제하기도 하고, \$B를 대치하여 "'ㄱ'의 뜻으로 바른 것은?"을 출제하기도 하는 알고리즘이다.

문제 유형은 최소화 할 수 있는 변수의 개수에 따라 유형 1, 2, 3, 4 ... 로 분리하기로 한다. 유형 1은 "예시+문제"로 최소단위 변수가 2개인 경우를 말한다. 유형 2는 "예시1+예시2+문제"로 최소단위 변수가 3개인 경우를 말한다. 유형 3은 "예시1+예시2+예시3+문제"로 최소단위 변수가 4개인 경우를 말한다. 유형4 이상의 문제 유형 또한 최소단위 변수의 개수에 따라 나눈다.

문제자동출제 알고리즘을 정리하면 다음과 같다. 첫째, 문제 자체를 유형별로 분석한다.

입력할 문제가 “예시+문제”의 형태인지, “문제+예시+문제”의 형태인지 등 어떤 형태로 구성되어 있는지를 분석한다. 이렇게 분석한 결과를 몇 개의 변수로 대입이 가능한지에 따라, 이미 만들어진 유형 중에 대입이 가능한지 분석한다. 위의 예시로 든 문제 “ex) ‘可’의 발음은?”는 ‘可’ 라는 예시와 ‘의 발음은?’ 이라는 문제 즉, “예시+문제”의 형식을 가진 유형에 해당하게 된다.

둘째, 분석한 문제를 세분화한다.

유형 분석이 끝난 문제는 구성요소들로 잘라낸다. 즉, 위에서 분석된 “可의 발음은?”는 “예시+문제”의 최소 단위로 세분화하여, 각각의 변수에 대입할 수 있게 준비한다.

셋째, 세분화한 데이터들을 문제은행에 입력한다.

세분화가 끝난 ‘문제’와 ‘문제 데이터’는 입력 폼을 통해 입력하게 된다. 입력 폼은 세분화된 문제와 문제 데이터의 최소단위 변수를 가지고 있는데, 이것은 문제 출제자가 문제 유형에 따라 임의로 지정할 수 있게 된다. 입력이 끝난 데이터는 지정된 변수의 규격에 맞추어 문제 은행에 저장된다.

넷째, 문제은행에 있는 데이터들을 조합하여 새로운 문제를 만든다.

저장된 ‘문제 데이터’와 ‘문제’를 문제은행으로부터 로드>Loading>하여, 이것을 ‘예시’는 ‘예시’별로, ‘문제’는 ‘문제’별로 대처하여, 하나의 새로운 문제를 만들어낸다. 이 때 ‘보기’와 ‘답’도 문제에 포함시키게 된다.

그러면 이런 과정을 통해 출제될 수 있는 문제의 개수를 알아본다.

문제의 데이터가 <표 1>과 같이 1개 입력되었고, 문제가 <표 2>와 같이 4개 입력되었다면, <표 3>와 같은 문제들이 자동출제 될 수 있다.

<표 1> 입력되어 있는 문제 데이터

可
---

<표 2> 입력되어있는 문제

[ ]를(을) 바르게 읽은 것은?
[ ]의 뜻으로 바른 것은?
[ ]의 활용으로 바른 것은?
[ ]와 가장 비슷한 뜻을 가진 한자는?

<표 3> 자동 출제되는 문제

"可"를(을) 바르게 읽은 것은?
"可"의 뜻으로 바른 것은?
"可"의 활용으로 바른 것은?
"可"와 가장 비슷한 뜻을 가진 한자는?

이렇게 총 4개의 문제가 만들어진다. 여기에 ‘人

’이라는 문제 데이터가 추가되었다고 한다면, <표 4>와 같은 문제들이 자동출제 될 수 있다.

<표 4> 문제데이터가 추가되었을 때 자동 출제되는 문제

"可"를(을) 바르게 읽은 것은?
"可"의 뜻으로 바른 것은?
"可"의 활용으로 바른 것은?
"可"와 가장 비슷한 뜻을 가진 한자는?
人"를(을) 바르게 읽은 것은?
人"의 뜻으로 바른 것은?
人"의 활용으로 바른 것은?
人"와 가장 비슷한 뜻을 가진 한자는?

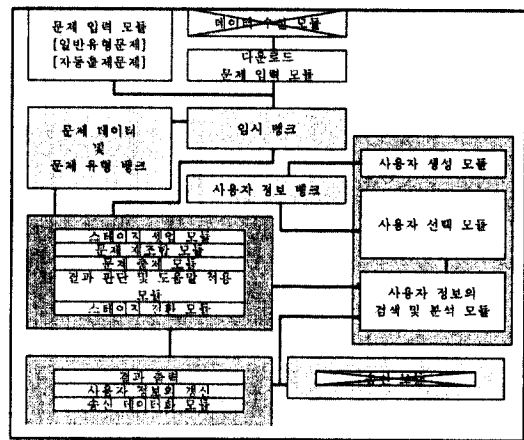
따라서, ‘문제데이터의 수 \* 문제의 수’ 만큼 문제는 만들어진다. 또한, ‘알파’로 여기에 답은 1~4번 사이에 무작위로 선정되며, 답을 제외한 보기 또한 입력된 데이터를 바탕으로 무작위하게 계속 변화한다.

## 2.2. 시스템의 설계 및 구현

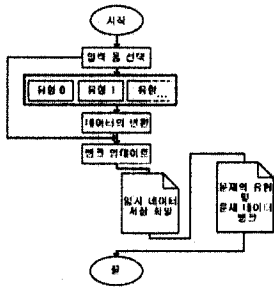
### 2.2.1 문제 입력과 학습 프로그램 설계

시스템을 위한 세부적인 구성 요소를 다음과 같이 설계하였다. 시스템의 전체 구성도는 (그림 1)과 같다. 시스템은 크게 두 부분으로 나누어 설계하였다. 첫째는 사용자가 문제들을 문제은행에 입력하는 부분이다. 둘째는 이렇게 입력된 문제데이터들을 이용하여 새로운 문제를 출제하고 학습하게 하는 부분이다[3, 4].

(그림 1) 시스템 구성도



가) 문제입력 프로그램

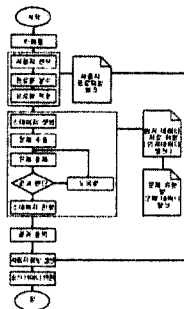


(그림 2) 문제입력프로그램 설계

문제입력 프로그램을 시작하면, 먼저 입력할 문제의 형식을 선택한다. 자동출제 알고리즘에 적용되지 않는 고정타입문제 유형, 자동출제 알고리즘에 적용되는 문제 유형, 타 학습자로부터 다운로드 받을 문제 유형 등이 있다. 이렇게 유형에 따라 입력한 문제 데이터는 बैं크(문제 은행)에 입력할 형식으로 바뀌어 बैं크에 입력된다. 이로써, 문제입력 프로그램은 종료된다(그림 2)[3, 4].

나) 문제자동출제에 의한 학습 프로그램

문제자동출제에 의한 학습프로그램을 시작하면, 먼저 타이틀화면이 뜬다. 사용자가 자신의 프로파일과 학습할 문제의 수를 선택하면, 성취도 및 난이도, 학습 진행률 등이 분석되어 학습스테이지가 구성된다. 문제를 풀어서 정답이면 다음 스테이지가 진행되고, 오답이면 도움말(힌트)이 나온다. 학습을 마치면, 학습자의 학습능력을 분석하여 학습결과를 출력하고 사용자의 프로파일을 갱신한다. 그리고, 사용자의 프로파일을 교육망으로 전송하기 위하여 데이터를 변환하여 기록한다(그림 3)[3, 4].



(그림 3) 문제자동출제에 의한 학습프로그램 설계

2.2.3 데이터 입력 및 관리

문제 입력 모듈은 문제은행에 문제를 입력하는 모듈이다. 고정타입문제, 자동출제문제, 타 학습자로부터 다운로드 받은 문제, 문제 데이터를 입력할 수 있다. 사용자는 먼저 (그림 4)을 이용하여 입력할 데이터의 형식을 결정하여야 한다[4].

(그림 4) 입력할 데이터의 형식 선택 창

가) 고정 타입 문제의 입력

고정타입문제는 문제자동출제 형식을 적용하기 힘든 문제이다. 문제, 답, 보기, 힌트 등을 하나의 데이터로 관리하여야 한다. (그림 5)와 같은 입력 폼으로 설계하였다.

(그림 5) 고정타입문제의 입력 폼

'NEW'를 클릭하면, 필드에 입력이 가능해진다. 모든 입력을 마친 후 'MAKE'를 클릭하면, 만들어진 문제가 화면에 디스플레이 되고, 최종 확인 절차를 거쳐서 문제 बैं크에 업데이트된다. 위쪽 중앙 부근의 스크롤 상자를 이용하여 문제유형 번호를 선택하면 다른 문제의 유형 폼으로 넘어 갈 수 있다[5-10].

나) 자동 출제 문제의 입력

자동출제문제의 입력은 문제의 유형을 등록할 수 있다. (그림 6)와 같은 입력 폼으로 설계하였고, 상단에 있는 도구 상자의 역할은 (그림 5)와 동일하다.

'NEW'를 클릭하면, 중앙 부분의 스크롤 상자 안에 유형 번호가 셋팅되고, 모든 입력 필드가 초기화된다. 입력을 모두 마친 후 왼쪽 상단의

'MAKE'를 클릭하면, 만들어진 문제가 화면에 디스플레이 되고, 최종 확인 절차를 거쳐서 문제뱅크에 업데이트된다[5-10].

(그림 6) 자동출제문제의 입력 폼

다) 문제 데이터의 입력

(그림 7)와 같은 입력 폼으로 설계하였다. NO SELECT 부분은 입력이 불가능한 필드를 나타낸다. 이것은 문제 유형에 따라 입력 필드의 개수가 결정 되기 때문인데, 여기서는 자동출제문제 유형 1번에 해당하는 필드의 예이다[5-10].

(그림 7) 문제 데이터의 입력

라) 다운로드 문제의 입력

교육망이나 메일을 통해 다운로드 받은 문제 데이터를 문제은행에 입력하기 위해, (그림 9)과 같은 입력 폼을 설계하였다.

'OPEN'을 클릭하면, 하드디스크에서 파일을 찾아 로드 할 수 있도록 (그림 8)과 같은 폼이 나타난다[5-10].

(그림 8) 다운로드 문제 파일 찾기

(그림 9) 다운로드 문제의 입력 폼

파일의 로드가 끝나면, 두 개의 필드 중 상단 필드에 로드한 데이터의 정보, 즉, 총 문제 수, 난이도, 만든 사람 등의 정보가 표시된다. 그리고 하단의 입력 폼에는 체크박스과 함께 문제 유형별 문제의 예제와 포함하고 있는 문제 수가 표시된다. 이것은 학습자가 원하는 것만 골라서 문제 은행 업데이트가 가능하도록 한다.

마) 문제 은행

문제은행은 문제를 유형별로 나누어 고정타입문제는 TYPE 0, 자동출제문제는 유형별로 TYPE 1, 2, ... 등으로 분류하여 저장한다.

이 때, 데이터의 인덱스 태그로 '[']를 사용한다. 인덱스 태그는 저장된 데이터의 분류를 알아야만 데이터 파싱이 가능하고 실제로 사용할 데이터로의 변환이 가능하기 때문에 필요하다. 이 인덱스 태그의 하단에 있는 내용들이 실제로 사용되는 데이터가 된다.

<표 5>는 문제/문제유형 은행에 저장되는 고정타입문제의 저장 형식이다. '<TYPE 0>'은 고정타입문제를 선언한다. '[문제]'는 문제의 내용을 나타내는 인덱스 태그이다. '[지문]'은 해당 문제의 보기를 나타내는 인덱스 태그이다. 위에서부터 1번, 2번, 3번, 4번의 보기를 기록한다. '[답]'은 문제의 답을 나타내는 인덱스 태그이다. '[힌트]'는 틀린 답을 클릭했을 때 표시할 힌트를 나타내는 인덱스 태그이다.

<표 5> 문제유형은행의 고정타입문제 저장 형식

```
<TYPE 0>
[문제]
'인'을 한자로 쓴 것 중 옳은 것은?
[지문]
愛人
憐人
人間
動物
[답]
1
[힌트]
사람 '애'는 '愛好' 등과 같은 말로도 사용됩니다.
[참]
```

지문(보기)의 데이터들로 지문을 따로 입력하지 않고 NULL로 셋팅하면, 답을 제외한 나머지 지문은 [NULL]에서 찾게 되는데, 이러한 보기의 저장 형식은 <표 6>와 같다.

<표 6> 문제유형은행의 NULL 보기 저장 형식

```
[NULL]
가, 나, 다, 라, 마, 바
[NULL]
家, 價, 可, 加, 假, 家
[NULL]
[NULL]
사랑하는 사람, "삼기 좋은 세상"
```

<표 7>은 문제/문제유형 은행에 저장되는 자동출제문제의 유형 저장 형식이다. '<TYPE 1>'은 자동출제문제 유형 1번임을 선언한다.

<표 7> 문제은행의 자동출제문제 유형의 저장 형식

```
<TYPE 1> A&B
<문제>
[한자](를) 바르게 읽은 것은?
[한자]의 뜻으로 바른것은?
[한자]의 활용 중 바른것은?
[한자]와(과) 가장 비슷한 뜻을 가진 한자는?
[뜻](뜻음)은 어떤 것인가?
<답>
[독음]
[뜻]
[활용]
[유사어]
[한자]
<지문>
NULL0
NULL1
NULL2
```

<표 8>는 문제/문제유형 은행에 저장되는 자동출제문제의 데이터 저장 형식이다. '<TYPE SET 1>'은 자동출제문제 유형 1번의 데이터라는 것을 선언한다.

<표 8> 문제 데이터의 저장 방식

```
<TYPE SET 1>
愛, 사랑, 애, 好, 愛人, 愛好
人, 사람, 인, 可, 商人, 商人
```

바) 임시 문제 은행

임시문제은행은 학습프로그램에 사용되는 서론, 해설, 도움말(힌트) 등을 보관한다. 그리고, 고정타

입문제, 자동출제를 위한 유형별 문제 데이터를 형식에 맞게 정의해 준다. 프로그램이 실행되는 동안, 학습자가 선택한 문제 개수와 난이도에 따른 스테이지 배정에 따라 수시로 갱신된다.

<표 9>는 임시문제은행에 인사말과 각 스테이지 도입 메시지가 저장되는 형식이다.

<표 9> 임시문제은행의 학습목표 저장 형식

```
[인사말]
안녕하세요. 제가 만든 문제를 풀어 보세요.
처음은 그렇게 힘든 문제가 나오지 않으니 두려워하실 것 없습니다.
[서론1]
제 1과입니다. 여기서는 독음에 대한 문제를 풀어 보도록 합니다.
[서론2]
제 2과입니다. 여기서는 한자의 뜻을 풀이해 보도록 합니다.
[서론3]
제 3과입니다. 여기서는 사자성에 대하여 공부해 보도록 합니다.
[서론4]
제 4과입니다. 훌륭하십니다. 벌써 마지막 과입니다. 파이팅!
```

<표 10>는 임시문제은행에 저장되는 고정타입 문제의 저장 형식이다. '[문제수]'는 사용자의 학습 정보 분석 모듈에서 입력한 문제의 수를 나타내는 인덱스 태그이다. '[문제]'는 문제의 내용을 나타내는 인덱스 태그이다. '[지문]'은 해당 문제의 보기를 나타내는 인덱스 태그이다. 위에서부터 1번, 2번, 3번, 4번의 보기를 기록한다. '[답]'은 문제의 답을 나타내는 인덱스 태그이다. '[힌트]'는 틀린 답을 클릭했을 때 표시할 힌트를 나타내는 인덱스 태그이다.

<표 10> 임시문제은행의 고정타입문제 저장 형식

```
[문제수]
5
[문제]
'愛人'을 뜻으로 풀이한 것 중 옳은 것은?
[지문]
사랑하는 사람
공부하는 사람
노력하는 사람
운명하는 사람
[답]
1
[힌트]
사람 '애'는 '愛好' 등과 같은 말로도 사용됩니다.
```

<표 11>은 임시문제은행에 저장되는 자동출제 문제의 저장 형식이다. '[문제]'는 자동출제문제 유형의 번호를 나타내는 인덱스 태그이다. 이 번호를 통해 문제은행을 참고할 수 있다. '[답]'은 문제의 답을 나타내는 인덱스 태그이다.

<표 11> 임시문제은행의 자동출제문제 저장 형식

```
[문제]
TYPE 1
```

사) 사용자 학습정보 은행

사용자의 학습정보 은행은 학습자의 성취도, 레벨 등을 저장하는 은행이다. <표 12>과 같은 형태로 저장된다.

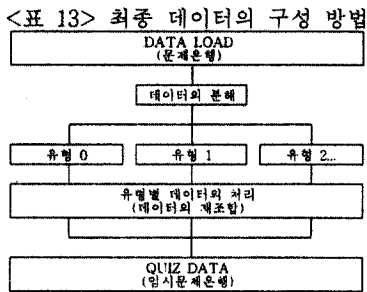
**<표 12> 사용자 학습정보 은행**

<입진회>	// 사용자의 이름
[학습 시간]	// 사용자의 최종 학습 시간을 나타내는 인덱스
2001.08.11 오후 9시 30분	
2001.08.12 오후 7시 33분	
2001.10.23 오후 11시 15분	
[출제 문제]	// 사용자가 학습한 문제의 학습 시간 별 기록
30	
16	
40	
76	
[총 출제 수]	// 사용자가 지금까지 학습한 문제의 총 개수
37.5%	
[정답률]	// 출제된 문제에 대한 정답률
90%	
[성취도]	// 정답률 90%이상 '수', . . . 50%이하 '가'
우	
[레벨]	// 다른 학습자와의 비교(랭킹)
출제회	
[강한 문제 유형]	// 가장 높은 정답률을 보이는 문제의 유형
TYPE 1	
[취약 문제 유형]	// 가장 낮은 정답률을 보이는 문제의 유형
TYPE 2	

2.2.4 재조합과 출제

<표 13>은 문제 재조합에 필요한 일련의 과정을 도식적으로 설명한 것이다.

이러한 과정 후에 학습자가 학습할 수 있는 최종 데이터가 만들어지며, 이 데이터를 스테이지에 출력하여, 학습자가 비로소 학습을 할 수 있게 된다. 문제 재조합 모듈은 각 유형별로 문제 유형과 문제 데이터를 재조합하는 프로세서를 두어 유형별로 문제를 재조합하고 임시 저장 은행에 저장한다. 문제 출제 모듈은 임시문제은행에 기록된 문제를 화면에 출력하는 모듈이다[3-10].



2.3 문제 유형별 시나리오

입력 모듈과 재조합 모듈, 그리고 출제의 과정을 모두 포함하여, 각각의 유형별로 정리해 보았다.

유형은 문제에 포함되어 있는 변수의 개수에

따라 분류될 수 있다.

이렇게 변수의 개수에 따라 나뉘어진 문제유형의 문제데이터들은 문제은행에 저장된다. 문제은행에는 문제유형의 문제데이터 뿐만 아니라 각각의 유형에 따라 만들어진 문제도 함께 저장된다.

2.3.1 문제 유형 0

문제 유형 0은 고정 타입 문제 유형을 말한다.

데이터 입력의 각각의 필드들은 고유의 변수명을 가지고 있고, 입력된 데이터는 이 변수에 1:1로 대입된다. 유형 0번의 문제의 경우 문제, 답, 보기 1, 보기 2, 보기 3, 힌트의 6개의 변수에 입력된 데이터들이 대입되어 있다. 이것은 data parse에 의하여 <표 14>과 같이 변수에 대입된다.

<표 14> 문제 유형 0의 변수 대입

SA = '문제'
SB = '답'
SC = '보기 1', SD = '보기 2', SE = '보기 3'
SF = '힌트'

이렇게 대입된 문제의 데이터는 유형별 데이터 처리 프로세서에 등록되어 있는 문제 유형에 따라 변수들을 재조합하게 된다. 문제 유형 0(고정 타입 문제)의 경우, 별다른 조합이 없이 문제 데이터를 그대로 사용하게 된다.

2.3.2 문제 유형 1

문제 유형 1은 '예시+문제'의 자동출제문제 유형이다.

유형 1번은 (그림 6)의 문제입력과 (그림 7)의 문제 데이터로 입력된 데이터를 <표 15>와 같이 세분화하여 변수에 대입한다.

<표 15> 유형1 문제 데이터의 변수 대입

SDA = '예'
SDB = '사랑'
SDC = '예'
SDD = '강'
SDE = '예나'
SDF = '예강'
SIIA = '사랑 예(예)는 예(예)에도강(예)도 합용 된다.'
SIIB = '예(예)은 나라를 사랑한다는 의미이다.'

이렇게 대입된 변수들은 문제의 유형에서 사용하게 된다. 유형별 데이터 처리 프로세서에서 이러한 변수를 사용할 규칙을 설정하고, 이것을 적용한다. 입력 폼으로 입력받아 은행에 저장된 문제의 데이터는 <표 16>과 같이 대입되어 불러진다.

<표 16> 유형 1 문제의 변수 대입

\$QA=[한자]을(를) 바르게 읽은 것은? \$QB=[한자]의 뜻으로 바른것은? \$QC=[한자]의 활용 중 바른것은? \$QD=[한자]와(과) 가장 비슷한 뜻을 가진 한자는? \$QE=[뜻][독음]는 어떤 것인가?  \$AA=[독음] \$AB=[뜻] \$AC=[활용] \$AD=[유사어] \$AE=[한자]
--

유형별 데이터 처리 프로세서에서는 이것을 적용할 규칙을 가지고 있다. 문제 유형 1번의 경우 기본 알고리즘이 "A+B"에 맞춰 있다. A는 "예시", B는 "문제"를 말한다. 즉 유형이 예시를 들고 질문하는 형식의 문제를 말한다.

<표 17> 유형 1 문제의 규칙

$$SD + SQ = SA$$

즉 <표 17>의 규칙에 맞게 <표 18>과 같은 문제가 만들어진다.

<표 18> 문제 출제 예

{\$DA + \$QA = \$AA} 문제 = '뜻'을(를) 바르게 읽은 것은? 답 = 예
--

(그림 10) 유형 1 문제 출력 화면

같은 유형의 다른 문제들도 위의 규칙을 따라가게 되며, 규칙에 따라 문제를 생성하게 된다. 생성된 데이터는 임시 뱅크에 기록된다. 이렇게 만들어진 문제는 (그림 10)과 같이 화면에 출력된다.

### 2.3.3 문제 유형 2

문제 유형 2은 '예시1+예시2+문제'의 자동출제문제 유형이다.

문제 유형 2번은 <"有備無患"은(는) "미리 준비해 두면 근심 될 것이 없음"이라는 의미의 사자성어이다. 다음중 의미가 비슷한 사자성어는?> 이것을 유형 2번으로 선택하여 설명한다. 이 유형은 <표 19>과 같이 나눌 수 있다.

<표 19> 유형 2 문제의 세분화

A = 有備無患 B = 은(는) C = 미리 준비해 두면 근심 될 것이 없음 D = 이라는 의미의 사자성어이다. E = 다음중 의미가 비슷한 사자성어는?
---

이렇게 나누어진 문제는 각각 "A, B, C, D, E"라 칭한다면, "A&B&C&D&E"라는 규칙으로 만들 수 있다. 이것을 자동 출제의 규칙에 맞추기 위하여 "문제"와 "문제 데이터"로 나누면, "B, D, E"는 문제가 되며, "A, C"는 문제 데이터가 된다.

이것은 입력 폼(그림 11)을 통하여 입력되는 형태는 <표 20>와 같다. 문제는 "(그림 11)에서 문제 데이터 부분은 (그림 12)에서 입력한다.

(그림 11) 유형 2 문제 입력

(그림 12) 유형 2 문제 데이터 입력

<표 20> 유형 2 문제의 예

[사자성어]은(는) [풀이] 이라는 의미의 사자성어이다 다음중 의미가 비슷한 사자성어는?
---

문제에 포함되어있는 "[ ]"는 문제 데이터를 표시하는 것으로 변수와 같은 존재이다.

이렇게 입력된 데이터는 문제은행에서 읽어 드린 후에 <표 21>의 형태로 변수에 대입한다.

<표 21> 유형 2 문제의 변수 대입

\$QA=[사자성어]은(는) [풀이] 이라는 의미의 사자성어이다 다음중 의미가 비슷한 사자성어는?  \$AA=[유사어]
--

이것은 "(그림 6) 문제 데이터의 입력"의 오른쪽 입력 필드의 이름으로 문제 유형 2번의 문제 데이터 입력을 할 경우 오른쪽 필드는 "(그림 12) 유형



2 문제 데이터 입력"과 같이 변화하고, 문제 데이터는 형식에 맞춰 입력한다. 이렇게 입력된 문제 데이터는 문제은행에 저장되어 있는 데이터를 읽어 들여 <표 22>의 구조로 변수에 대입된다.

<표 22> 유형 2 문제 데이터의 변수 대입

\$DA = '有兩無惡'
\$DB = '유비무환'
\$DC = '미리 준비해 두면 근심 될 것이 없음'
\$DD = '狡獪三窟'
\$DE = '교토삼굴'
\$DF = '피 많은 토끼가 굴을 세 개 가지고 있기에 죽음을 면할 수 있었다'
\$HA = '편안할 때에 위기를 생각하십시오(居安思危). 그러면 대비를 하게 되며(思患而豫備), 대비태세가 되어 있으면 근심이 사라지게 됩니다(有備無患).'

이렇게 해서 만들어지는 문제의 수는 "입력 된 문제 데이터 \* 문제의 수"로 유형 1과 동일하다. 이것은 (그림 13) 와 같이 출력된다.

(그림 13) 유형 2 문제 출력 화면

이렇게 입력 된 데이터는 같은 데이터를 활용하는 같은 유형의 문제를 <표 23>과 같이 만들어 내어 적용 할 수 있다.

<표 23> 유형 2 문제

"有兩無惡"은(는) "미리 준비해 두면 근심 될 것이 없음"이라는 의미의 사자성어이다. 다음 중 이 사자성어의 독음으로 바뀐 것은? "有兩無惡"은(는) "미리 준비해 두면 근심 될 것이 없음"이라는 의미의 사자성어이다. 다음 뜻들이 중 의미가 같은 것은? "미리 준비해 두면 근심 될 것이 없음"이라는 뜻을 가진 사자성어로 "狡獪三窟"이 있다. 다음 중 의미가 같은 한자어는? "미리 준비해 두면 근심 될 것이 없음"이라는 뜻을 가진 사자성어로 "狡獪三窟"이 있다. 다음 중 의미가 같은 한자의 독음은?
--

2.3.4 문제 유형 3

문제 유형 3은 '예시1+예시2+예시3+문제'의 자동 출제문제 유형이다. <"釋迦塔", "多寶塔", "石窟庵">들과 관계있는 한자어는?>과 같은 문제가 있을 수 있다. 이것을 문제의 최소 단위로 세분화한 것을 각각 "A, B, C, D"라 칭한다면, "[A&B&C]&D"라는 규칙으로 만들 수 있다. 이것을 자동 출제의 규칙에 맞추기 위하여 "문제"와 "문제 데이터"로 나누면, "D"는 문제가 되며, "A, B, C"는 문제 데이터가 된다<표 24>.

<표 24> 문제 유형 3의 세분화

A = 釋迦塔 B = 多寶塔 C = 石窟庵 D =들과 관계있는 한자어는?
---

<표 25>은 문제를 입력하고 이를 변수에 대입했을 때의 모습이다.

<표 25> 유형 3 문제의 변수 대입

\$QA=[관련어1][관련어2][관련어3]들과 관계있는 한자어는? \$AA=[관련어4]
---

<표 26>은 문제 데이터를 입력하여 문제 은행에 저장한 데이터를 읽어 들인 후 이를 변수에 대입한 상태를 표시하는 것이다.

<표 26> 유형 3 문제 데이터의 변수 대입

\$DA = '釋迦塔' \$DB = '多寶塔' \$DC = '石窟庵' \$DD = '佛國寺' \$HA = '호국불교(高國佛敎)는 나라를 보호한다는 불교이다.'
--

2.3.5 유형 4 이상의 문제

그 밖의 유형 문제 또한 위에서 설명한 방식으로 최소 단위로 세분화하여, 재조합하는 방식을 활용하여 생성해 나간다.

2.4 프로그램의 응용분야 및 한계점

이상의 프로그램의 개발로 다른 과목에 적용을 시도해 본다.

외국어 분야에 대해선 한자와 같은 원리로 적용이 가능하다. 기본 원리는 모두 동일하므로 설명은 생략하고, 적용이 가능한 문제들을 설명해 본다. 한자에 맞춘 본 프로그램을 영어와 일본어에 적용하면, 문제 유형 1번은 <표 27>와 같은 문제, 문제 유형 2번은 <표 28>와 같은 문제, 문제 유형 3번은 <표 29>과 같은 문제가 가능하다.

<표 27> 영어, 일어의 유형 1 적용 예

"Love"의 뜻으로 맞는 것은? "Love"와 비슷한 뜻을 가진 단어는? "あい"의 뜻으로 맞는 것은? "あい"와 비슷한 뜻을 가진 단어는?
--

<표 28> 영어, 일어의 유형 2 적용 예

"Love"는 "사랑"이라는 뜻을 가지고 있다. 이와 유사한 뜻을 가진 단어는? "あい"은(는) "사랑"이라는 의미의 단어다. 이와 유사한 뜻을 가진 단어는?
---

<표 29> 영어, 일어의 유형 3 적용 예

( ) You는 "나는 당신을 사랑합니다."라는 뜻이다. 괄호에 들어갈 말로 맞는 것은?  
 I ( ) You는 "나는 당신을 사랑합니다."라는 뜻이다. 괄호에 들어갈 말로 맞는 것은?  
 I Love ( )는 "나는 당신을 사랑합니다."라는 뜻이다. 괄호에 들어갈 말로 맞는 것은?  
 ( )は あなたを あいしてるとよ"는 "나는 당신을 사랑합니다."라는 뜻이다. 괄호에 들어갈 말로 맞는 것은?  
 "わたしは ( )を あいしてるとよ"는 "나는 당신을 사랑합니다."라는 뜻이다. 괄호에 들어갈 말로 맞는 것은?  
 "わたしは あなたを ( ) いてるとよ"는 "나는 당신을 사랑합니다."라는 뜻이다. 괄호에 들어갈 말로 맞는 것은?

이와 같이 영어, 일본어는 물론이고, 독일어, 러시아어, 불어 등 외국어 영역에 대한 간단한 문제에 적용이 가능하다.

그러면, 다른 영역의 과목에 대하여 살펴보자. 제한적이지만 수학에도 일부 적용이 가능하다. 예를 들면, 문제 유형 1번은 <표 30>와 같은 문제, 문제 유형 2번은 <표 31>와 같은 문제가 가능하다. 문제 유형 3번의 경우도 적용이 가능하다.

<표 30> 수학의 유형 1 적용 예

"1"의 값은?  
 "1"의 값과 같은 값이 나오는 수식은?

<표 31> 수학의 유형 2 적용 예

"1"과 "2\*3"을 각각 계산 한 후 이를 서로 더한 값은?  
 "1"과 "2\*3"을 각각 계산 한 후 이를 서로 곱한 값은?  
 "1"과 "2\*3"을 각각 계산한다. 이 둘은 같은 어떤 것인가?

본 프로그램(문제자동출제시스템)은 예로 든 문제의 유형처럼 문제를 최소 단위로 세분화 할 수 있는 간단한 문제에만 적용이 가능하다. 보다 수준 높은 문제를 출제자, 교수자의 의도대로 출제하고자 한다면, 이러한 자동 출제는 의미가 없어진다. 즉, 특수한 문제의 경우에는 문제유형 0번(고정 타입 문제)를 활용하여야 한다.

문제를 최소 단위로 세분화하여 재조합하는 알고리즘의 자동출제시스템은 문제를 최소단위로 나눌 수 없는 경우가 발생한다면, 이것을 유형으로 만들어 문제 데이터화하는 것이 불가능하였다.

3. 결론

교육부가 [7차 교육과정]을 본격적으로 실행하면서, 기존의 교과서와 칠판을 교육의 도구로 활용해 왔던 교육방식이, 컴퓨터를 교육매체로 활용하는 멀티미디어활용교육방식으로 빠르게 전환되고 있다[1, 2]. 이러한 상황에 요구되는 학습 프로그램의 개발이 시급하다는데 초점을 맞추어 연구하게 되었고, 다음과 같은 효과를 기대할 수 있었다.

첫째, 학습자는 게임을 즐기듯이 학습용 프로그램을 통해 학습하게 되어, 보다 흥미를 가지고 학

습할 수 있게 되었다. 더불어 편안하고 친근한 디자인의 클라이언트는 누구나 쉽게 다가갈 수 있게 만들어졌다.

둘째, 학습자는 문제자동출제 알고리즘에 의해 계속 생성되는 문제 속에서 다양하고 보다 폭넓은 학습효과를 얻을 수 있게 되었으며, 적은 노력으로 많은 문제를 만들어내게 되어 데이터의 업데이트에 있어 인력, 재정적인 절감 효과를 볼 수 있게 되었다.

셋째, 학습자는 스스로 문제를 만드는 과정에서 한번 학습을 하게 되고, 재창조된 문제를 풀어가면서 다시 한번 복습하는 효과를 얻을 수 있었다. 이러한 과정 속에서 학습자는 보다 높은 학습 효과를 얻을 수 있게 되었다.

넷째, 학습자는 타 학습자의 데이터를 받아 학습할 수 있게 됨에 따라, 보다 다양한 학습 정보를 접할 수 있게 되었으며, 교수자→학습자의 수직적이고 일방적인 구조의 네트워크에서 벗어나 학습자↔학습자, 교수자↔학습자의 수평적이고 상호보완적인 네트워크를 구성할 수 있게 되었다. 이를 통한 학습 커뮤니티가 생성되고, 학습자-학습자간의 경쟁심과 협동심이 동시에 발생하여, 학습자간의 시너지 효과를 볼 수 있게 되었다.

참고 문헌

- [1] 교육부(2001). - 교육정보화촉진시행계획
- [2] 교육부(1998). - 제 7차 교육과정은 이렇게 구성되었습니다 - 자율과 창의에 바탕을 둔 학생 중심 교육과정.
- [3] AndrewRollings, Dave Morris.(2000). - Game Architecture And Design - The Coriolis Group. All Rights Reserved.
- [4]. Kyle Loudon (2000). - Algorithms with C
- [5] Microsoft Visual C++ 6.0 ATL COM(2000). - 삼양출판사
- [6] George Shephed and Scot Wingo(1999). - MFC Internals - ADDISON-WESLEY
- [7] Beginning MFC COM Programming(1997). - Wrox Press Ltd.
- [8] C++ Programming LANGUAGE THIRD EDITION(1997). - ADDISON-WESLEY, AT&T.
- [9] C++ FAQs SECOND EDITION(1999). -

ADDISON-WESLEY

- [10] GAME Programming GEMS(2000). - CHARLES RIVER MEDIA, INC.
- [11] ICT교육의 현황 - <http://www.etimesi.com>  
긴급점검-초중고 정보화교육
- [12] 에듀넷 - <http://www.edunet4u.net>
- [13] 김경아(2000) - 웹 기반 자동 문제 출제 시스템 - 동국대학교 교육대학원 컴퓨터교육전공
- [14] 김대인(1998) - WWW에서 문제은행 중심의 교육용 코스웨어 개발을 위한 공동저작 시스템의 설계 및 구현 - 원광대학교 교육대학원 전자계산 교육전공
- [15] 엄혜진(1998) - 문제중심학습의 상호작용 증진을 위한 온라인학습 플랫폼 프로토타입 개발 - 한양대학교 대학원 교육공학과전공