

## 트리즈를 활용한 소프트웨어 개발에 관한 연구

김은경\* · 구본철

### A Study of Applying TRIZ in Software Development

Eun-Gyung Kim\* · Bon-Chul Koo

School of Computer Science & Engineering, Korea University of Technology and Education, Cheonan  
330-708, Korea

#### 요 약

창의적 문제해결 이론인 트리즈는 기계나 전기, 전자 등과 같은 전통적인 공학 분야에서 널리 활용되어 좋은 성과를 거두고 있으나, 아직 소프트웨어 분야에 적용된 사례는 많지 않다. 하지만 소프트웨어도 다른 어떤 분야 못지않게 창의적 문제해결 역량이 요구되는 분야이므로, 트리즈를 활용한다면 놀랄 만한 성과를 얻을 수 있을 것으로 예상된다. 따라서 본 논문에서는 트리즈를 적용해서 스마트 클릭러라는 새로운 소프트웨어를 개발해 봄으로써 소프트웨어 분야에서는 트리즈를 활용하기 어렵다는 심리적 타성을 타파하고, 소프트웨어 분야에서 트리즈를 활용하는 접근방법을 제시하였다.

#### ABSTRACT

TRIZ, the theory of inventive problem solving, is widely used in traditional engineering fields such as mechanical, electrical, and electronic engineering and generated good results. Yet there are few examples in software field. But software field requires creative problem solving capabilities more than any other fields and we expect that we can get remarkable results by applying TRIZ in this field. In this paper, we try to break the psychological inertia that TRIZ is difficult to use in software area through development of new software, the Smart Clicker, by applying TRIZ. And we present an approach how to apply TRIZ in software field.

**키워드** : 트리즈, 시스템 사고, 이상적 최종 결과, 기술적 모순, 스마트 클릭러

**Key word** : TRIZ, System Thinking, IFR, Technical Contradiction, Smart Clicker

접수일자 : 2013. 12. 09 심사완료일자 : 2014. 01. 08 게재확정일자 : 2014. 01. 22

\* **Corresponding Author** Eun-Gyung Kim(E-mail:egkim@kut.ac.kr, Tel:+82-41-560-1350)

School of Computer Science & Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

**Open Access** <http://dx.doi.org/10.6109/jkiice.2014.18.3.719>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.  
Copyright © The Korea Institute of Information and Communication Engineering.

## I. 서 론

트리즈(TRIZ)는 창의적 문제해결 이론(Theory of Inventive Problem Solving)이라는 의미를 갖는 러시아어의 첫 글자를 딴 약어로서, 러시아의 과학자이자 기술자인 겐리히 알트슐러를 중심으로 전 세계 수백만 건의 특허 분석을 기초로 정립된 이론이다[1-6]. 트리즈는 창의적이고 효율적인 방법으로 문제를 해결할 수 있도록 도와주는 다양한 도구와 방법을 제공하는 매우 체계적이고 구조화된 접근 방법이지만, 원래 물리나 화학, 기계, 전자공학과 같은 전통적인 과학 및 공학적 원리에 기초하고 있어서, 주로 이들 분야에서 활용되어 좋은 성과를 거두어 왔다. 최근에는 반도체나 디스플레이 분야 등 다양한 IT 분야에서도 트리즈를 적용한 괄목할 만한 성과가 많이 보고되고 있으며, 비즈니스 분야나 다양한 사회적 갈등을 해결하는 문제에도 적용하려는 시도가 확산되고 있는 추세이다. 그러나 소프트웨어 분야에서 트리즈를 활용하는 방법에 대해서는 아직 명확한 방향이 제시되지 못하고 있는 실정이다.

소프트웨어 분야는 다른 어떤 분야 못지않게 창의적 역량이 요구되는 분야로서, 트리즈를 활용할 수 있는 접점을 찾는다면 그 어느 분야보다 괄목할만한 성과를 얻을 수 있을 것으로 기대된다. 따라서 본 논문에서는 트리즈를 활용해서 새로운 소프트웨어를 개발해 봄으로써, 소프트웨어 분야에서는 트리즈를 활용하기 어렵다는 심리적 타성을 타파하고, 한 발 더 나아가 소프트웨어 분야에서 트리즈를 효과적으로 활용할 수 있는 방안을 제시하고자 한다.

## II. 트리즈의 주요 개념

트리즈는 매우 다양한 도구와 이론을 포함하는 있으나, 본 연구에서는 다음과 같은 몇 가지 도구와 이론을 활용하였다.

### 2.1. 시스템 사고(System Thinking)

시스템 사고란 문제를 하나의 시스템으로 이해하고, 그림 1과 같이 9개의 창을 통해서 보다 넓은 시야에서 문제를 파악하고 문제 해결 방향을 모색하는 사고 기법이다[1,4]. 즉, 어떤 문제가 발생한 시스템 자체에서만 고려

하는 것이 아니라, 시스템을 구성하는 모든 구성 요소나 부품에 해당하는 하위 시스템이나 시스템과 상호작용하거나 시스템에 영향을 미칠 수 있는 모든 것에 해당하는 상위 시스템도 함께 고려하는 것이다. 또한, 현재 시점에서만 문제를 찾거나 해결 방향을 모색하는 것이 아니라, 과거나 미래도 함께 고려하는 사고 기법이다.

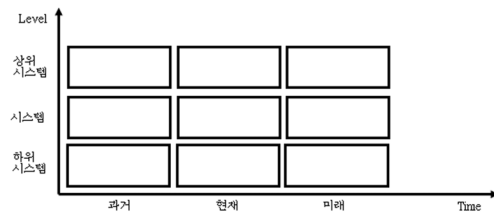


그림 1. 시스템 사고를 위한 9개의 창  
Fig. 1 9 Windows for System Thinking

### 2.2. IFR과 이상성

IFR(Ideal Final Result: 이상적 최종 결과)은 바람직한 결과가 스스로 달성되는 것으로, 문제의 궁극적인 이상적 해결안을 의미한다. 트리즈의 기술시스템 진화의 법칙 가운데 하나인 이상성 증가의 법칙에 의하면, 모든 시스템은 이상성을 증가하는 방향으로 진화하는데, 이상성 정도 D(Degree of Ideality)는 수식 (1)과 같이 평가한다[1-4].

$$D = \frac{\sum \text{유리한 효과}}{\sum \text{비용}} \quad (1)$$

수식 (1)에서 분모의 비용은 경제적인 비용뿐만 아니라 유해한 효과도 포함하며, 가장 이상적인 시스템은 D가 무한대가 되는 것으로, 즉 비용이 0이 되는 시스템이라고 할 수 있다. 이상성을 증가시킬 수 있는 방법은 비용을 감소시키거나 유용한 효과를 증가시키는 것이며, 궁극적으로 시스템의 기능을 외부나 상위시스템으로 전이시킴으로써 기능은 존재하지만 시스템은 존재하지 않는 상태가 가장 이상적인 시스템이다.

### 2.3. 기술적 모순과 발명 원리

트리즈의 기술적 모순(Technical Contradiction: TC)은 기술 시스템이 내포하는 두 개의 특성이 서로 충돌하는 상황을 의미한다[1,4,5,6]. 예를 들면, 자동차에 출력이 큰 엔진을 장착하면 승차감은 좋아지지만 연비가

나빠지고, 반대로 출력이 작은 엔진을 장착하면 연비는 좋아지지만 승차감은 나빠지는데, 이처럼 승차감이라는 특성과 연비라는 특성이 서로 충돌하는 상황을 기술적 모순이라고 한다.

트리즈는 기술적 모순을 근본적으로 해결할 수 있는 40가지 발명원리(Inventive Principles)를 제공하며, 어떤 두 특성이 상호 충돌할 때 어떤 발명원리를 적용하는 것이 효과적인지 쉽게 판단할 수 있는 모순행렬(Contradiction Matrix)을 제공하고 있는데, 이에 대한 자세한 설명은 참고문헌으로 대신하겠다[1,2]. 모순 행렬을 활용하기 위해서는 먼저 개선되는 특성과 악화되는 특성에 해당하는 변수를 결정해야 하는데, 알트슐러가 정의한 39개의 공학변수나 최근에 새로 정의된 비즈니스 변수나 소프트웨어 변수를 활용할 수 있다. 본 연구에서는 인벤션머신(Invention Machine)사의 이삭복만 등이 개발한 소프트웨어용 모순행렬과 소프트웨어 변수를 활용하였다.

### III. 클릭어의 문제해결 방향 모색

많은 학생들이 교수의 질문에 응답을 잘 하지 않고, 적극적으로 질문하는 학생도 많지 않은 것이 현실이다. 이런 문제를 해결하기 위해서 예일대에서 클릭어(Clicker)라는 개인응답시스템(Personal Response System: PRS)[7,8]을 개발하게 되었는데, 미국의 경우 액티브 러닝(Active Learning)을 위한 하나의 도구로써 클릭어를 많이 활용하고 있으며, 최근 국내에서도 클릭어의 활용이 증가하고 있다. 클릭어는 일반적으로 응답용 리모컨과 수신기 및 응답 통계 처리용 소프트웨어로 구성되어 있는데, 구입 및 유지보수 비용이 부담이 되는 것이 현실이다. 또한 일반적인 클릭어는 응답 및 집계 기능은 있지만 질문 기능이 없다는 한계점이 있다. 본 연구에서는 트리즈를 적용해서 현 클릭어의 문제를 해결할 수 있는 바람직한 방향을 모색하였다.

#### 3.1. 시스템 사고를 통한 문제해결 방향 모색

현재 문제가 되고 있는 시스템인 클릭어를 중심으로 표 1과 같이 9개의 창을 통해서 전체적인 모습을 파악하면, 보다 바람직한 문제 해결 방향 즉, 새로운 문제를 인식할 수 있다. 현재 시스템인 클릭어를 개선하는 문

제를 인식할 수도 있지만, 클릭어를 대체할 수 있는 미래 시스템이 무엇인지를 예측하는 것이 더 바람직하다. 본 연구에서는 클릭어의 미래 시스템은 응답뿐만 아니라 질문도 가능하고 별도의 수신기 없이 응답 수신기 가능한 시스템으로, 이를 스마트 클릭어(Smart Clicker)라 칭하였다.

표 1. 클릭어에 대한 시스템 사고  
Table. 1 System Thinking for the Clicker

구분	과거	현재	미래
상위 시스템	학생, 교사, 수업, 강의실	학생, 교사, 수업, 강의실, 라디오주파수	학생(스마트폰), 교사, 수업, 블루투스 통신
시스템	O/X 혹은 숫자 카드	클릭어	스마트 클릭어
하위 시스템	종이, 잉크	리모컨(응답용), 수신기, SW	리모컨(응답/질문용), 통신모듈, SW

#### 3.2. IFR을 이용한 문제해결 방향 모색

IFR을 이용해서 문제를 해결하기 위해서는 먼저 주어진 문제 상황을 분석해서 문제가 무엇인지 정확히 정의한 다음, 문제 해결에 활용할 수 있는 자원(Resource)이 무엇인지 찾아보거나, 시스템 스스로 문제를 해결할 수 있는 이상적인 방법을 모색한다.

##### 3.2.1. 문제(Disadvantage) 정의

현재 사용되는 일반적인 클릭어는 다음과 같은 문제 즉, 단점이 있다.

- ① 학생 수 만큼의 리모컨을 구입해야 한다.
- ② 응답 수신기와 집계 프로그램을 구입해야 한다.
- ③ 질문 기능이 없다.
- ④ 유지보수 비용이 필요하다.

##### 3.2.2. 문제를 해결할 수 있는 자원

클릭어의 내외부 자원 가운데 위의 문제를 해결하는데 활용할 수 있는 자원은 무엇일까? 그 자원은 다음의 두 가지로 요약할 수 있다.

- ① 물질(Substance): 리모컨 대신 스마트폰(거의 모든 학생들이 소지하고 있는 자원) 활용
- ② 정보(Information): 무선통신을 통해서 수신기 없이도 응답 수신기 가능함

### 3.2.3. 시스템 스스로 문제를 해결하는 방법

클리커 자체가 위의 문제들을 스스로 해결하려면 어떻게 해야 하나? 앞에서 언급한 바와 같이 가장 이상적 시스템은 기능은 존재하지만 시스템은 존재하지 않는 것이다. 즉, 클리커 없이도 응답, 질의, 응답통계 기능을 수행할 수 있는 방법이 가장 이상적이며, 트리즈의 이상성 증가의 법칙에 의하면 기능을 상위 시스템으로 전이해서 이상성을 증가시킬 수 있다. 표 1에서 알 수 있듯이 클리커의 상위 시스템 가운데 하나는 학생이며, 넓게는 학생이 항상 소지하는 스마트폰도 상위 시스템이 될 수 있다. 따라서 스마트폰에 클리커의 기능을 전이시킨다면, 별도로 클리커를 구입할 필요가 없으므로 구입이나 유지보수 문제가 자동으로 해결되고, 질문 기능도 쉽게 구현할 수 있으므로 시스템 스스로 위의 문제를 해결하게 된다.

결된 곳이면 어디에서나 등록할 수 있다. 다음으로 강의용 PC에 MFC로 개발된 클라이언트 프로그램을 설치해야 한다. 클라이언트 프로그램은 학생들이 응답한 결과를 실시간으로 집계하여 교수용 PC 하단에 표시하고, 설정된 응답 시간이 종료되면 최종 집계 결과를 PC 화면에 표시해서 교수와 학생 상호간의 즉각적인 피드백이 가능하도록 구현하였다. 또, 학생이 텍스트로 질문한 내용도 PC 하단에 실시간 표시해준다.

## IV. 클리커 설계 및 구현

시스템 사고와 IFR을 활용해서 다각도로 문제해결 방향을 모색한 결과, 클리커의 상위 시스템인 스마트폰에 클리커의 기능을 전이하는 것이 가장 바람직한 문제해결 방향임을 확인하였다. 따라서 본 논문에서는 기존 클리커의 리모컨을 대체할 수 있도록 응답 기능 외에 질문 기능을 추가한 AOS(Android Operating System) 스마트폰의 앱(App)과 응답집계 결과 및 학생 질문을 교수용 PC에 실시간 표시하는 MFC 프로그램, 그리고 다양한 유형의 통계 처리 및 질문 등록 등이 가능한 웹 프로그램으로 구성된 스마트 클리커를 개발하였다.

### 4.1. 스마트 클리커의 구성

스마트 클리커는 그림 2와 같이 크게 웹서버와 교수용 PC 그리고 스마트폰 앱으로 구성된다. 본 연구에서 구현한 스마트 클리커는 한 장소에서 일회성으로 사용하는 것이 아니라, 한 대학에서 여러 교수가 여러 강의에서 활용할 수 있도록 구현하였다. 또 한 강의에서 클리커를 사용해서 질의, 응답한 통계를 다양한 형태로 표시하는 기능도 제공하는데, 이를 위해서 통합서버인 APM 서버를 구축하고 PHP로 웹 페이지를 개발하였다. 교수가 스마트 클리커를 사용하려면 먼저 웹 페이지에 로그인해서 해당 강의를 등록해야 하는데, 인터넷이 연

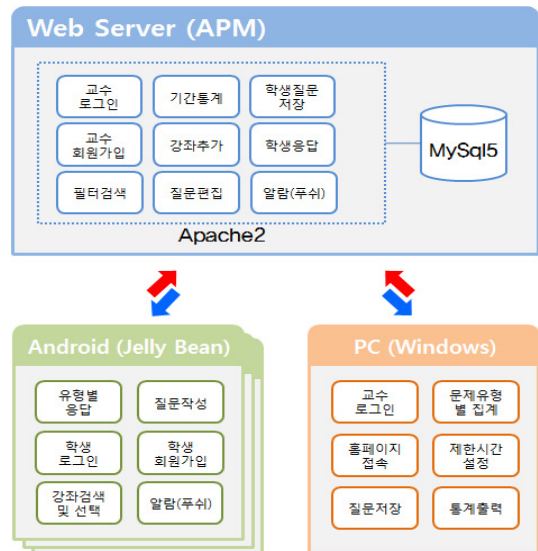


그림 2. 스마트 클리커의 구성도  
Fig. 2 Configuration of the Smart Clicker

학생이 스마트 클리커를 사용하려면 먼저 마켓에서 스마트 클리커 앱을 다운로드 받아 계정을 생성한 다음, 앱을 실행해서 “설정“ 버튼을 터치하면 스마트 클리커를 사용하는 강좌목록이 표시되는데, 목록에서 자신이 수강하는 강의를 선택해야 한다. 학생은 교수의 질문에 O/X나 숫자 또는 단답형으로 답할 수 있으며, 텍스트로 질문하는 것도 가능하다.

### 4.2. 스마트 클리커 구현 시의 모순 해결

소프트웨어를 개발하면서 직면하는 많은 문제가 모순 상황이다. 예를 들면, 데이터 저장을 위해 배열(Array)로 구현하면 구현이 쉽다는 장점은 있지만, 데이터가 계속 추가되는 경우 대처하기 어렵다는 단점이 있고, 반대로 연결 리스트(Linked List)로 구현하면 데이터가 추가되

어도 쉽게 대처할 수 있다는 장점이 있지만, 구현이 복잡하다는 단점이 있다. 개발자가 원하는 것은 구현도 간단하면서 데이터가 동적으로 증가해도 쉽게 대처할 수 있는 자료구조이지만, 이 두 가지를 동시에 만족시키는 자료구조는 현재로서는 없다. 소프트웨어 개발 시에 이런 모순에 직면하게 되면 대개 주어진 상황에서 어떤 장점이 더 중요한지를 판단해서 한 가지를 선택하게 되는데, 이 경우 보다 효율적인 방법으로 구현했다고는 볼 수 있으나, 트리즈 관점에서 모순이 해결된 궁극적인 해결안이라고 보기 어렵다. 본 연구에서 스마트 클릭어를 구현함에 있어서도 여러 모순 상황에 직면하게 되었는데, 다음 사례와 같이 모순 해결을 통해서 궁극적인 해결안을 도출하여 구현하였다.

4.2.1. 서버와 스마트폰 앱의 세션 연결 문제

클릭어의 핵심 기능이라고 할 수 있는 응답 기능을 구현할 때 서버와의 세션 연결 문제가 발생한다. 즉, 수업 중 교수의 질문에 즉각 응답하기 위해서는 스마트폰 앱과 서버 간의 세션이 연결되어 있어야 하는데, 사실 언제 질문할지 모르는 상황에서 세션을 계속 연결해두는 것은 효율적이지 않다.

(1) 기술적 모순 정의

위의 문제 상황에 포함된 기술적 모순(TC)은 표 2와 같이 두 가지로 정의할 수 있다.

표 2. 세션 연결의 기술적 모순  
Table. 2 Technical Cont. of the Session Connection

TC1	세션을 계속 연결해 두면, 교수의 질문에 즉각 응답할 수 있다는 장점이 있지만, 서버 자원 활용이 비효율적이라는 단점이 있다.
TC2	세션을 끊어 두면, 서버 자원의 활용이 효율적이라는 장점이 있지만, 교수의 질문에 즉각 응답하기 어렵다는 단점이 있다.

(2) 진짜 문제 정의

기술적 모순 분석을 기초로 진짜 문제(Real Problem)를 정의하기 위해서는 표 2의 두 가지 기술적 모순 가운데 하나를 선택해야 한다. 즉 세션을 계속 연결해 두면서 서버 자원의 활용을 효율화하는 방법을 모색하거나, 세션을 끊어 두면서 질문에 즉각 응답할 수 있는 방법을 모색해서 모순을 해결해야 한다. 클릭어의 핵심 기

능인 즉각 응답 기능이 중요하므로, 본 연구에서는 TC1을 선택하여 다음과 같이 진짜 문제를 정의하였다.

스마트폰 앱과 서버 간의 세션을 연결해 두면서 서버 자원을 효율적으로 활용할 수 있는 방법을 모색하자.

(3) 모순행렬 활용을 위한 변수 결정

TC1에서 개선되는 특성은 즉각 응답할 수 있다는 것으로, 이는 시간 손실을 줄이는 것으로 볼 수도 있으므로 12번 ‘시간 손실’이라는 변수를 선택하였고, 악화되는 특성은 세션을 계속 연결해두어 서버의 메모리 및 저장 공간이라는 자원을 더 많이 소비하는 것이므로 19번 ‘자원 소비’ 변수를 선택하였다.

표 3. 세션 연결의 TC1에 대한 소프트웨어 변수  
Table. 3 Software Variables for TC1 of Session Connection

구분	개선되는 특성	악화되는 특성
특성	즉각 응답	자원
변수	시간 손실(12번)	자원 소비(19번)

(4) 발명원리 탐색 및 아이디어 도출

개선되는 특성 12번과 악화되는 특성 19번을 소프트웨어 모순행렬에 대입하면 발명원리 10, 20, 21번을 탐색할 수 있는데, 본 논문에서는 이 가운데 20번 ‘유용한 작용 지속 원리’를 활용하여 표 4와 같은 문제해결 방향을 결정하였다.

표 4. 유용한 작용 지속 원리를 활용한 아이디어 도출  
Table. 4 Idea Generation using Continuity of Useful Action Principle

개념적인 해결안	구체적인 해결안
자원이 낭비되지 않으면서 유용한 작용인 세션 연결이 유지되도록 함	세션을 계속 연결해두되 서버의 자원 낭비를 최소화하기 위해서 스마트 클릭어 앱이 백그라운드에서 수행되도록 구현하고, 응답해야 하는 시점을 푸시(Push) 알림 방식으로 알려주도록 구현함

(5) 아이디어 구현

그림 3은 교수의 질문에 대해 학생이 응답해야 함을 푸시 알림 방식으로 알려주는 화면으로, 스마트폰 화면에 표시된 알림 메시지를 클릭하면 뒤의 그림 6과 같이 학생이 응답할 수 있는 화면이 표시된다.

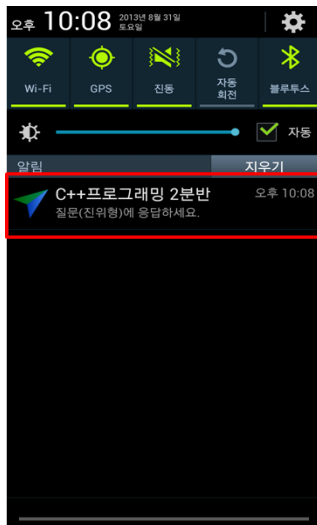


그림 3. 푸쉬로 응답 시점을 알려주는 화면  
 Fig. 3. Screen of the Response Time Notification by Push

4.2.2. 강의목록 다운로드 문제

앞에서 설명한 바와 같이 학생이 스마트 클릭기를 사용하기 위해서는 먼저 스마트 클릭기를 사용하는 전체 강의 목록 가운데 자신이 수강하는 강의를 선택해야 한다. 그런데 학기 중에 스마트 클릭기를 사용하는 강의가 수시로 추가될 수 있으므로 강의목록은 계속 바뀔 수 있다. 따라서 강의목록을 서버에 저장해서 계속 최신화시킬 필요가 있는데, 그렇게 되면 학생들의 스마트폰으로 매번 수정된 강의목록을 다운로드 해야 하고, 또 강의목록을 검색할 때 매번 서버에 접속해야 하므로 데이터 사용량이 증가하는 문제가 발생한다. 만약 학생이 선택한 강의목록만 스마트폰에 저장하면 데이터 사용량이 증가하는 문제는 해결되지만, 강의목록이 최신화되지 않아 학생이 수강하는 강의가 목록에 새로 추가되어도 학생이 이를 인지할 수 없게 된다.

(1) 기술적 모순 정의

위의 문제 상황에 포함된 기술적 모순(TC)은 표 5와 같이 두 가지로 정의할 수 있다.

(2) 진짜 문제 정의

기술적 모순 분석을 기초로 진짜 문제를 정의하기 위해서는 표 5의 두 가지 기술적 모순 가운데 하나를 선택해야 한다.

표 5. 강의목록 다운로드의 기술적 모순

Table. 5 Technical Cont. of the Lecture List Download

TC1	강의목록을 서버에 저장하면, 목록 최신화가 용이하다는 장점이 있지만, 데이터 사용량이 증가하는 단점이 있다.
TC2	강의목록을 스마트폰에 저장하면, 데이터 사용량이 감소하는 장점이 있지만, 강의목록이 최신화되지 않는 단점이 있다.

스마트폰 사용자의 경우 데이터 사용량에 민감한 경우를 무시할 수 없으므로, 본 연구에서는 TC2를 선택하여 다음과 같이 진짜 문제를 정의하였다.

강의목록을 스마트폰에 저장해 두면서 강의목록을 최신화시킬 수 있는 방법을 모색하자.

(3) 모순행렬 활용을 위한 변수 결정

TC2에서 개선되는 특성은 데이터 사용량이 감소하는 것으로, 이는 자원 소비가 감소하는 것으로 볼 수 있으므로 19번 ‘자원 소비’ 변수를 선택하고, 악화되는 특성은 강의목록이 최신화되지 않는다는 것으로, 이는 데이터 무결성이 보장되지 않는 것이므로 6번 ‘데이터 무결함성’ 변수를 선택하였다.

표 6. 강의목록 다운로드의 TC2에 대한 소프트웨어 변수

Table. 6 Software Variables for TC2 of the Course List Download

구분	개선되는 특성	악화되는 특성
특성	데이터 사용량 감소	강의목록 최신화 불가
변수	자원소비 (19번)	데이터 무결함성 (6번)

(4) 발명원리 탐색 및 아이디어 도출

개선되는 특성 19번과 악화되는 특성 6번을 소프트웨어용 모순행렬에 대입하면 불행히도 발명원리를 탐색할 수 없다. 따라서 본 연구에서는 40가지 발명원리 가운데 3번 국소적 성질(Local Quality) 원리를 활용해서 표 7과 같은 문제해결 방향을 결정하였다.

(5) 아이디어 구현

그림 4는 로그인할 때 서버의 강좌목록이 갱신되었음을 알려주는 화면으로, ‘예’ 버튼을 터치하면 갱신된 강좌목록을 스마트폰으로 다운로드된다.

표 7. 국소적 성질 원리를 활용한 아이디어 도출  
Table. 7 Idea Generation using Local Quality Principle

개념적인 해결안	구체적인 해결안
전체 강의목록은 서버에 저장하고, 학생이 선택한 강의 목록만 스마트폰에 저장하도록 구현한다.	전체 강의목록은 서버에 저장하고, 학생이 선택한 강의목록은 스마트폰의 내부 DB에 저장하도록 구현하고, 서버의 강의목록이 갱신되면 스마트폰에 알려준다음, 학생이 승인하는 경우에만 갱신된 강의목록을 스마트폰에 새로 다운로드하도록 구현한다.

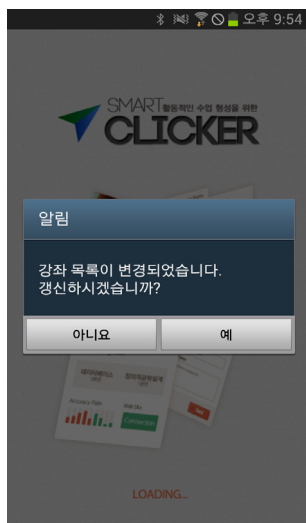


그림 4. 강좌목록 갱신을 알려주는 화면  
Fig. 4 Screen of the Courses List Update Notification

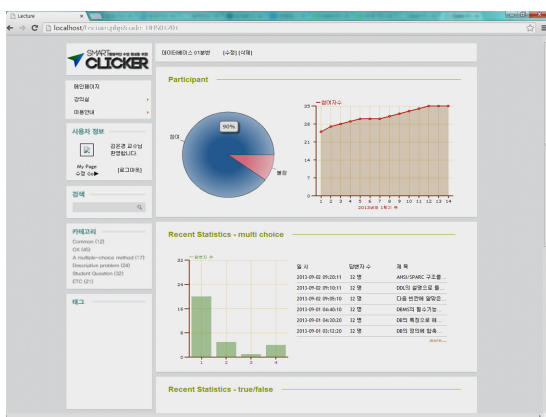


그림 5. 응답 통계 및 참여 표시 화면  
Fig. 5 Screen of the Response Statistics & Participation

### 4.3. 스마트 클릭어의 주요 실행 화면

그림 5는 교수용 웹 페이지에서 교수가 스마트 클릭어 웹페이지에 접속해서 어떤 한 강의에서 학생들이 응답한 통계와 참여율을 확인하는 화면이고, 그림 6은 학생용 스마트폰 앱에서 학생이 텍스트로 질문하는 화면이다.



그림 6. 텍스트로 질문하는 화면  
Fig. 6 Screen of a Text Question

그림 7은 교수용 클라이언트 프로그램에서 제한 시간이 종료된 다음 최종 집계 결과를 교수용 PC에 표시한 화면이다.

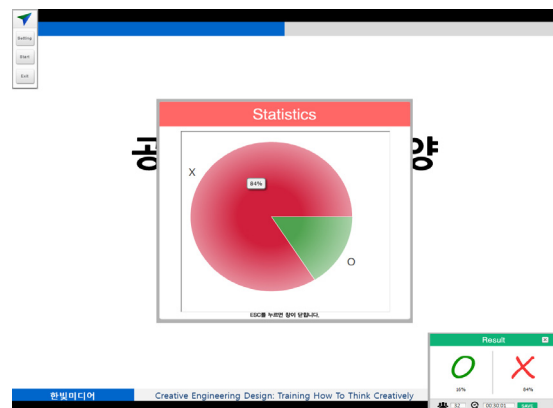


그림 7. 최종 집계 결과 표시 화면  
Fig. 7 Screen of a Final Aggregate Result



## V. 결론

모든 다른 개발 분야와 마찬가지로 소프트웨어 개발에 있어서도 가장 중요한 것은 문제 인식이다. 즉, 어떤 소프트웨어를 개발할 것인가를 결정하는 것이 무엇보다 중요하다. 이런 의미에서 트리즈를 활용해서 어떤 소프트웨어를 개발할 것인가에 대한 개발 방향을 모색하는 것은 매우 유의미한 것으로 판단된다. 본 연구에서는 트리즈의 시스템 사고와 IFR을 활용해서 개인응답시스템인 클릭커의 문제점을 분석한 결과, 기존 클릭커의 기능을 상위 시스템인 스마트폰으로 전이시키는 것이 매우 이상적인 문제임을 인식하였다.

또한, 스마트 클릭커를 구현할 때 당연한 여러 문제도 모순 분석을 기초로 해결 방향을 모색함으로써, 한 가지 장점을 선택하는 대신 다른 장점은 포기하는 것이 아니라, 각 선택의 장점을 모두 취할 수 있는 이상적인 해결안을 모색하여 구현하였다. 물론 트리즈를 사용하지 않아도 동일한 해결책으로 구현할 수 있겠지만, 본 연구를 통해서 소프트웨어 개발 시에 당연하게 되는 많은 문제를 트리즈를 활용해서 해결할 수 있음을 확인한 것은 매우 유의미하다고 판단된다. 본 연구에서는 소프트웨어 개발과 관련된 문제를 주로 기술적 모순 분석과 발명원리를 활용해서 해결했지만, 앞으로 보다 다양한 트리즈 기법을 적용해볼 계획이며, 이러한 연구를 통해서 소프트웨어 분야에서 트리즈를 보다

효율적으로 활용할 수 있는 보가 체계적인 방법을 제시할 계획이다.

## REFERENCES

- [1] Genrich Altshuller, *Creativity as an Exact Science - The Theory of Inventive Problem Solving*, CRC, 1984.
- [2] Genrich Altshuller, *Innovation Algorithm, Interscience*, 2006.
- [3] Semyon D. Savransky, *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving*, CRC Press, 2000.
- [4] EunGyung Kim, *Creative Engineering Design: Training How to Think Creatively*, Hanbit Media, 2013.
- [5] EunGyung Kim, "TRIZ-based Real Problem Definition Process," *KIPEE*, Journal of KIPEE, Vol. 2, No. 2, pp.28-34, 2009.
- [6] EunGyung Kim, "TRIZ-based Problem Definition Process for Creative Problem Solving," *ICEE*, Proceedings of ICEE, pp.391-397, 2009.
- [7] Michelle Hudson, Laurie McGowan, Cheri Smith, "Technology and Learner Motivation in Library Instruction: A Study of Personal Response System," *Indiana Libraries*, Vol. 30, Number 1, 2011.
- [8] Margie Martyn, "Clickers in the Classroom: An Active Learning Approach," *Educause Quarterly*, Number 2, pp. 71-74, 2007.



김은경 (Eun-Gyung Kim)

1983년 2월 : 숙명여자대학교 물리학과 졸업  
1986년 2월 : 중앙대학교 전자계산학과 석사  
1991년 2월 : 중앙대학교 컴퓨터공학과 박사  
1992년 3월 ~ 현재 : 한국기술교육대학교 컴퓨터공학부 교수  
※관심분야 : 스마트러닝, 트리즈, 지능형 에이전트 등



구본철(Bon-Chul Koo)

2013년 한국기술교육대학교 컴퓨터공학부 공학사  
2013년~현재 한국기술교육대학교 컴퓨터공학부 석사과정  
※관심분야 : 트리즈, 정보통신, 클라우드 컴퓨팅