Journal of Korea Robotics Society (2015) 10(2):096-103
http://dx.doi.org/10.7746/jkros.2015.10.2.096 ISSN: 1975-6291 / eISSN: 2287-3961

코딩 교육용 소프트웨어에 대한 단위 행동 손실률 기반의 사용성 평가 방법

A Usability Evaluation Method Based on the Lostness of Unit Action for Coding Education Software

박 희 성[1], 박 광 현†

Hee Sung Park[1], Kwang-Hyun Park†

Abstract In this paper, we first point out the limitation of the existing methods in usability evaluation, and propose a new method to find tasks and users that can cause problems in the use of software applications. To deal with object-based applications, we define unit actions and divide a given task into the sequence of unit actions. Then, we propose a new measure to calculate the degree of lostness for the unit actions. In several experiments, we show the proposed method can represent the usability well while the previous method has a problem in object-based applications. We also find that the user's evaluation is more related to the proposed method than the previous method based on execution time through correlation analysis.

Keywords: Usability Evaluation, Lostness, Unit Action, Coding Education

1. 서 론

최근 IT 기술의 급격한 발전으로 PC 또는 모바일 환경에서 사용되는 다양한 애플리케이션이 개발되고 있으며, 사용자 경험(UX; User Experience)에 대한 중요성도 높아지고 있다[1]. 특히 교육용 로봇 분야[2,3]에서 아이들을 위한 교육용 애플리케이션의 경우, 사용법을 익히는데 많은 시간을 소비하지 않고 교육 목적에 맞게 애플리케이션을 사용할 수 있는지를 분석하는 것은 매우 중요하다.

사용성 평가는 UX의 하위 분야로서, 특정 환경에서 특정 작업을 수행하는 사용자들이 제품 및 시스템을 평가하는데 사용되는 방법이다. 사용성 평가는 크게 두 가지로 분류되는데, 평가자가 직접 시스템의 사용성을 평가하는 검사 방법(Inspection Methods)과 사용자가 시스템을 사용하는 것을 관찰하여 평가하는 검증 방법(Test Methods)이 있다. 검사 방법에는 전문가 평가(Heuristic Evaluation), 인지적 시찰법(Cognitive Walkthrough), 행동 분석(Action Analysis) 등이 있으며, 검증 방법에는 소리 내어 말하기(Thinking Aloud), 현장 조사(Field Observation), 설문 조사(Questionnaires) 등이 있다. 평가 방법이 다양한 만큼 평가자는 평가 목적과 상황에 맞는 평가 방법을 빠르게 선택함과 동시에, 적은 노력과 비용으로 제품 및 시스템을 평가하는 것이 중요하다[4]. 하지만 사용성 평가를 시행할 때 표 1과 같이 평가 방법에 맞추어 전문적인 지식과 특정 장비가 필요하거나 많은 시간 동안 사용자가 함께 있어야 하는 문제가 있다.

이를 해결하기 위한 기존의 방법으로는 시스템을 사용하는 시간을 측정하여 시간 데이터를 기반으로 문제가 되는 작업 및 사용자를 선별하는 방법, 자동화 분석 도구를 사용하여 사용 시간 측정과 설문 결과를 자동으로 분석하는 방법, 시스템을 사용하는 사용자의 행동 수를 측정하여 손실률을 통해 문제가 되는 작업 및 사용자를 선별하는 방법이 있다. 하지만 기존의 방법들은 사용자가 수행하는 작업의 시간과 성공 여부를 객관적으로 측정할 수 없거나,

Received : Apr. 13. 2015; Reviewed : Apr. 20. 2015; Accpeted : Apr. 29. 2015
† Corresponding author: School of Robotics, Kwangwoon University, Wolgye-Dong, Nowon-Gu, Seoul, Korea (akaii@kw.ac.kr)
1 Control and Instrumentation Engineering, Kwangwoon University (parkhs0602@lycos.co.kr)

Table 1. Usability Evaluation Methods

	Inspection Methods			Test Methods		
	Heuristic Evaluation	Cognitive Walkthrough	Action Analysis	Thinking Aloud	Field Observation	Questionnaires
Application Step	All	All	Design	Design	Final	All
Required Time	Low	Medium	High	High	Medium	Low
Required Users	None	None	None	3+	20+	30+
Required Evaluators	3+	3+	1-2	1	1+	1
Required Equipment	Low	Low	Low	High	Medium	Low
Required Expertise	Medium	High	High	Medium	High	Low
Involvement of Evaluator	No	No	No	Yes	Yes	No

웹 페이지에만 적용되는 한계점이 있다. 본 논문에서는 이러한 문제점들을 보완하기 위하여 PC 또는 모바일 환경에서 스크래치, 스택 미니 등과 같은 객체 기반의 애플리케이션에 대해 단위 행동 기반의 손실률을 제안하고 이를 기반으로 문제가 되는 작업 및 사용자를 선별하는 방법을 다룬다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 사용성 평가 방법에 대한 문제점을 살펴본다. 3장에서는 기존 방법의 한계점을 보완하기 위하여 단위 행동을 정의하고 손실률을 제안한다. 4장에서는 제안한 방법에 대한 타당성을 검증하기 위한 실험을 수행하고, 5장에서 결론을 정리한다.

2. 사용성 평가 방법[5-7]

평가자가 직접 시스템의 사용성을 평가하는 검사 방법은 실제로 제품이 어떤 과정으로 사용되는지를 파악하고 사용자의 요구를 이해하기 위한 것으로, 전문가 평가, 인지적 시찰법, 행동 분석 등이 있다. 전문가 평가는 전문가들이 사용성 원칙을 기준으로 대상이 되는 제품이 그 원칙에 부합하는 정도를 평가하고 문제점을 발견하여 평가하는 방법이다. 인지적 시찰법은 한 명 또는 여러 명의 평가자들이 종이 목업, 작동 가능한 시제품, 또는 이미 개발된 제품을 대상으로 기능 및 인터페이스가 사용하기 쉬운지 또는 어느 정도 이해할 수 있는지에 대해 평가자가 직접 작업을 수행하면서 조사하는 것이다. 행동 분석은 평가자가 주어진 작업을 수행하는 일련의 행동 과정을 분석하는 방법이며, 마우스의 움직임, 아이 트래커를 이용한 동공의 크기와 움직임, 얼굴 표정 등을 분석한다.

사용자가 시스템을 사용하는 것을 관찰하여 평가하는 검증 방법으로는 소리 내어 말하기, 현장 조사, 설문 조사 등이 있다. 소리 내어 말하기는 주어진 작업을 수행하는 과정에서 의식 또는 무의식적으로 일어나는 일련의 사고의 흐름을 말로 표현하여 평가자에게 알리는 방법이다. 현장 조사는 평가자가 실제 제품 및 시스템을 사용하는 사용자의 환경에서 사용자를 평가하는 방법이다. 설문 조사는 평가자가 미리 준비한 질문 목록을 사용자에게 나누어 주고 답변을 받는 방법이다.

사용성 평가를 시행할 때 평가 방법에 맞추어 전문적인 지식과 특정 장비가 필요하거나 많은 시간 사용자가 함께 있어야 하는 문제점을 해결하기 위한 기존의 연구로는 시간 측정 연구, 자동화 분석 연구, 행동 측정 연구가 있다.

2.1 시간 측정 연구

시간 측정은 측정한 시간 데이터를 통해 문제가 되는 작업과 사용자를 선별하는 방법으로, 참여자가 작업을 더 빠르게 완료할수록 제품이나 시스템의 효율성이 높다고 판단한다. 작업의 시작부터 완료할 때까지 걸린 시간은 진행자 또는 기록자가 스톱워치나 기타 다른 시간 기록 장치를 이용하여 분과 초 단위로 측정하거나, 실험자가 작업을 수행한 영상을 보고 측정한다. 하지만 시간 데이터에는 작업 수행의 시작과 끝을 정의하기 어렵다는 한계점이 있다. 시간을 어떻게 측정할 것인지에 대해서도 규칙이 필요하다. 무엇보다도 시계를 언제 켜고 끌 것인지가 가장 중요하다. 시계를 켜는 것은 참여자가 수행할 작업을 큰 소리로 읽도록 하여 읽는 것이 끝나자마자 시계를 켜는 것으로 작업의 시작점을 정할 수 있다. 하지만 시계를 끄는 것은 상당한 오류가 포함된다. 참여자들이 제품과 상호작용하는 것을 완료하였을 때 시간 측정을 중지해야 하는데, 관찰자 또는 평가자의 주관적인 판단이 폭넓게 관여되기 때문에 데이터에 상당한 오류가 포함될 수 있다. 참여자의

행동에도 오류가 발생할 수 있다. 참여자들이 작업을 수행하는 도중에 다른 행동을 하여 데이터가 오염될 수 있고, 참여자들이 같은 행동을 하더라도 참여자마다 차이가 발생한다. 평가자는 데이터를 분석할 때 이러한 오류를 제외해야 하는데, 평가자마다 오류를 제외하는 기준이 다를 수 있기 때문에 평가자의 주관적인 판단이 폭넓게 관여된다[8].

2.3. 자동화 분석 연구

데이터 로거(Data Logger) 또는 UTE(Usability Testing Environment) 등의 자동화 분석 도구는 표 2와 같이 사용자가 제품 또는 시스템을 사용할 때 설문 조사 결과 및 작업 수행 시간을 수집하여 설문 조사 점수, 작업 성공 여부, 작업 수행의 평균 시간 등을 표와 그래프로 나타내 준다[9]. 하지만 사용자에게 프로그램의 버튼을 통해 작업 수행 시간을 기록하도록 하기 때문에 사용자가 버튼을 누르지 않고 작업을 수행하는 경우 정확한 시간이 측정되기 어렵다. 또한 추가적으로 자동화 분석 도구를 실행하여야 하기 때문에 인위적인 환경에서 측정하게 되며, 작업의 성공 여부를 사용자의 주관에 맡겨 평가하기 때문에 객관적인 결과보다는 주관적인 측정 결과가 나오게 된다.

Table 2. Data Logger and UTE

	Data Logger	UTE
Type (Application Scope)	Excel Program (All)	Installation Program (Web)
Data Acquisition	Questionnaires, Task Execution Time	Questionnaires, Task Execution Time
Output	Questionnaire Results, Task Success Rate, Average Execution Time	Questionnaire Results, Average Execution Time

2.4. 행동 측정 연구

웹 형태 연구에서는 손실률을 측정하여 문제가 되는 작업 및 사용자를 선별한다[10]. 손실률 L은 식 (1)과 같이 사용자가 웹을 방문한 수를 통해 계산한다.

$$L = \sqrt{\left(\frac{N}{S}-1\right)^2 + \left(\frac{R}{N}-1\right)^2} \quad (1)$$

여기서, N은 작업을 수행하는 동안 방문한 서로 다른 웹 페이지의 개수, S는 작업을 수행하는 동안 방문한 전체 페이지의 개수(동일한 페이지를 다시 방문한 횟수도 포함), R은 작업을 수행하기 위해 반드시 방문해야 하는 페이지의 최소 개수를 의미한다.

사용자가 웹 페이지를 방문할 때 얼마나 중복된 페이지와 다른 페이지를 방문하였는지를 측정하여 손실률을 계산하는데, 손실률이 0.4 이하인 참여자들에게서는 관찰할 만한 특징이 나타나지 않으며 손실률이 0.5 이상인 참여자들에게서는 관찰할 만한 특징이 명확하게 나타난다는 사실을 발견하였다.

하지만, 식 (1)의 손실률은 페이지 기반으로 구성된 웹 페이지에서만 적용이 가능하다는 한계가 있다. 예를 들어 그림 1과 그림 2의 스크래치 프로그램[11]과 같이 객체 기반의 애플리케이션에서는 기존의 방법으로 손실률을 계산할 때 문제점이 나타난다. 그림 1과 같이 고양이를 앞으로 10만큼 움직이는 블록을 작업 영역의 어느 부분에 놓는지에 따라 같은 행동이지만 다른 행동으로 간주되기 때문이다. 또한 그림 2와 같이 고양이가 “Hello”라고 두 번 말하는

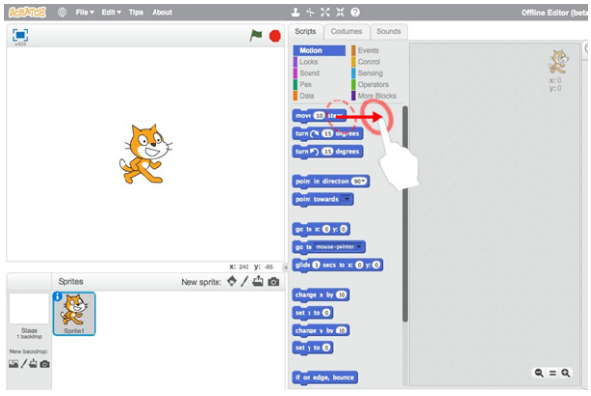

Fig. 1. Object-Based Application

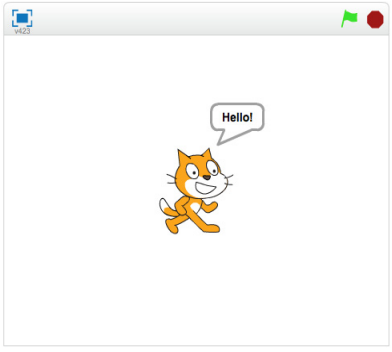

Fig. 2. Say “Hello” Twice

작업이 주어졌을 때, 중복된 행동이라도 옳게 수행한 작업일 수가 있다.

3. 단위 행동 기반의 손실률

기존의 방법 중에서 시간 측정 연구는 데이터가 오염될 수 있고, 자동화 도구 연구는 작업의 성공을 객관적으로 판별할 수 없으며, 웹 기반의 행동 측정 연구는 적용 범위가 제한적이라는 한계가 있다. 이를 해결하기 위하여 사용자 평가 시 데이터의 오염이 없고, 작업의 성공을 객관적으로 판별할 수 있으며, 객체 기반의 애플리케이션에 적용할 수 있는 방법을 제안한다.

본 연구에서는 사용자가 수행하는 작업이 서로 다른 행동으로 수행되지 않는다고 가정한다. 예를 들어 폴더를 삭제하는 작업을 수행하기 위한 행동은 폴더를 클릭한 후 키보드에서 삭제(Delete) 키를 누를 수도 있고, 폴더를 클릭한 후 마우스 오른쪽 버튼을 눌러 삭제 메뉴를 선택할 수도 있다. 이와 같이 하나의 작업을 서로 다른 행동으로 수행하는 경우에는 최소의 행동 수를 정의하기 어렵기 때문에 이러한 작업은 평가에서 제외한다. 본 연구에서는 작업 자체를 평가하는 것이 아니라 제품 또는 시스템의 사용성을 평가하는 것이기 때문에 이러한 가정에 맞게 작업을 설계할 수 있다. 따라서, 제안하는 방법의 적용 범위를 제한하는 것이 아니며 타당한 가정으로 볼 수 있다.

기존 연구의 한계점을 보완하기 위한 방법으로, 우선 작업의 행동 수를 측정할 수 있는 단위 행동을 정의하고, 객체 기반의 애플리케이션에 적용할 수 있는 손실률을 제안한다.

3.1 단위 행동

단위 행동은 더 이상 분리할 수 없는 최소의 행동 단위를 말한다. 사용성 평가에서 문제가 되는 작업을 선별하는 경우에는 작업의 단위가 커서 문제 작업을 선별하더라도 분석해야 하는 범위가 넓을 수 있다. 따라서 작업을 구성하는 단위 행동을 기준으로 문제가 되는 행동을 선별하면 분석 범위를 좁힐 수 있는 장점이 있다.

본 연구에서는 단위 행동을 행동 객체 이름과 입력 제스처의 쌍으로 구성한다. 행동 객체 이름은 사용자가 작업을 수행하는 대상이 되는 객체를 말하며, 그림 3은 행동 객체의 예시로서 스크래치 프로그램의 블록을 나타낸다. 입력 제스처는 사용자가 수행한 제스처를 나타내는데, 작업을 수행하는 환경이 PC 환경이면 마우스와 키보드의 입력이 제스처가 되고, 모바일 환경이면 그림 4와 같이 화면을 터치하는 제스처가 입력 제스처가 된다. 예를 들어 탭(화면을 한 번 터치하는 동작), 더블 탭(화면을 두 번 터치하는 동작), 드래그(화면을 터치하여 미는 동작) 등이 있다[12]. 표 3은 행동 객체 이름과 입력 제스처를 조합하여 단위 행동을 구성한 예시를 보여 준다.

Fig. 3. Blocks in Scratch Program

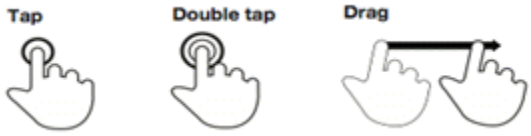

Fig. 4. Input Gestures in Mobile Environment

Table 3. Examples of Unit Action

Unit Action	Meaning
(move ~ steps, tap)	Touch once the "move ~ steps" block
(move ~ steps, drag)	Drag the "move ~ steps" block

3.2 단위 행동 기반의 손실률

객체 기반의 애플리케이션에 적용하기 위하여 식 (2)와 같은 손실률을 제안한다.

$$L = \frac{E}{S}, E = S - R \tag{2}$$

여기서 S는 작업 수행 시 사용자가 수행한 전체 단위 행동의 수, R은 작업을 수행하기 위한 최소의 단위 행동 수를 나타내며, E는 작업 수행 시 사용자가 수행한 잘못된 단위 행동의 수를 의미한다.

손실률 L은 작업을 위해 사용자가 수행한 전체 행동 수 S와 잘못된 행동 수 E의 비율로 계산하는데, 이를 통해 문제가 되는 단위 행동 및 사용자를 선별할 수 있다.

4. 실험 및 결과

4.1 실험 환경

기존의 손실률 방법의 문제점을 확인하고 제안한 손실률 방법의 타당성을 검증하기 위하여 '스택 미니' 애플리케이션을 대상으로 실험을 수행하였다. 스택 미니는 그림 5와 같이 모바일 환경에서 '알버트' 로봇을 활용하여 아이들을 대상으로 코딩 교육을 수행하기 위한 애플리케이션이다.

실험은 이러한 종류의 애플리케이션(스크래치 및 스택 미니 등)을 사용해 본 적이 없는 사용자 30명을 대상으로 하였으며, 스택 미니의 여러 가지 미션 중 '안녕, 알버트' 미션을 수행하도록 하였다. 이 미션은 표 4와 같이 12개의 작업으로 구성되어 있다. 사용자가 작업을 수행하는 과정을 영상으로 녹화한 후, 단위 행동을 분석하여 손실률을 계산하였다.

[Fig 5] Stack Mini Application

Table 4. 12 Tasks

No	Task
1	Say hello
2	Move forward twice
3	Move backward four times
4	Turn to the left and right
5	Move forward four times
6	Repeat moving forward and backward two times
7	Repeat moving backward and forward two times
8	Turn eyes on and move forward three times
9	Move forward three times and turn eyes on
10	Turn eyes on, move forward three times, turn to the right, and move forward four times
11	Repeat turning to the left until finding a hand
12	Repeat moving forward and backward until finding a hand

4.2 기존 손실률 방법의 문제점 검증

녹화된 영상을 분석하여 사용자 및 작업 별로 사용자가 수행한 단위 행동의 목록과 각 단위 행동의 시작 시간 및 종료 시간을 구하였다.

기존의 손실률 방법은 페이지 기반이기 때문에 사용자의 단위 행동을 하나의 페이지로 간주하여 계산하였다. 즉, 식 (1)에서 N은 작업을 수행하는 동안 보인 다른 행동의 수, S는 전체 행동 수, R은 작업에 필요한 최소 행동 수에 대응한다.

작업 3(알버트가 네 번 뒤로 이동하기)에 대한 최소의 단위 행동은 (도전하기, 탭), (뒤로 이동하기, 드래그), (뒤로 이동하기, 드래그), (뒤로 이동하기, 드래그), (뒤로 이동하기, 드래그), (프로그램 실행, 탭), (다음 단계로, 탭)으로 구성되며 R 값은 7이 된다. 사용자가 실제 수행한 단위 행동은 앞서 나열한 최소 단위 행동과 같았는데, 기존 방법에 의하면 다른 단위 행동의 수 N은 4이고 전체 단위 행동의 수 S는 7이므로, 손실률 식 (1)의 결과는 약 0.86으로 큰 손실을 보인다. 하지만, 제안한 손실률 방법에서는 전체 단위 행동의 수 S가 7, 최소의 단위 행동의 수 R이 7이므로, 손실률 식 (2)의 결과는 0이 되어 전혀 손실(잘못된 행동) 없이 작업을 수행한 것으로 나타난다. 실제로 사용자가 작업 3을 수행한 영상을 살펴보았을 때 전혀 문제 없이 작업을 수행하고 있음을 관찰할 수 있었는데, 기존 방법으로는 0.86이라는 높은 손실률 값이 계산되어 오류가 발생하였다. 즉, 문제 없이 작업을 수행한 사용자에게 잘못된 행동을 하였다는 결과가 나온 것이다. 이는 기존의 방법이 (뒤로 이동하기, 드래그)라는 중복된 단위 행동을 손실로 간주하기 때문이다.

4.3 시간 측정 연구와 제안한 방법의 비교

기존 방법 중에는 사용자가 작업을 수행한 시간을 측정하여 문제가 되는 작업 및 사용자를 선별하는 방법이 있

다. 즉, 시간이 오래 걸리는 경우 문제가 있다고 판단하는 것이다.

실험을 통해 측정된 시간과 제안한 손실률을 직접 비교하기는 불가능하기 때문에 사용자의 사후 평가 점수를 기준으로 비교하였다. 또한 사용자가 모든 단위 행동에 대해 사후 평가를 하는 것은 어렵기 때문에 촬영된 영상을 분석하여 가장 많이 나타나는 잘못된 행동을 5개의 그룹으로 분류하여 사후 평가 문항을 구성하였으며(표 5), 30명의 사용자가 수행한 작업을 촬영한 영상을 관찰하면서 각 문항에 대해 1점부터 5점까지 점수를 부여하도록 하였다.

본 실험은 단위 행동을 수행하는데 걸린 시간과 잘못된 단위 행동의 수 중에서 어느 것이 더 실제 평가와 관련이 높은지 알아보기 위한 것으로 다음과 같은 가설을 검증하기 위한 것이다.

가설: 사용자의 사후 평가 점수는 수행 시간보다 잘못된 행동의 수와 관련이 높다.

Table 5. Items for Post Evaluation

Item	Details
I cannot understand a given task	- See helps over and over - Use unnecessary blocks - Retry a given mission many times
I have difficulty in moving blocks	- Make a mistake in moving blocks
I cannot understand the functionality of blocks	- Use unnecessary blocks - Use blocks in a wrong way
I make a mistake in the use of graphic buttons	- Touch a picture - Touch unnecessary icons
I make a great effort to find a block	- Drag workspace over and over

사용자가 작업을 수행하면서 잘못 수행하였다고 느끼는 부분이 단위 행동의 시간과 잘못된 단위 행동의 수 중에서 어느 것과 더 관련이 높은지를 알아보기 위해 상관 분석을 하였다. 일반적으로 두 대상의 관련성을 확인하기 위해 공분산을 사용하는데, 공분산은 두 변수의 측정 단위에 따라서 차이가 발생하는 문제점이 있어 좋은 지표가 되지 못한다. 즉 본 실험에서는 시간과 개수, 사후 평가 점수의 물리적 단위가 다르기 때문에 공분산으로 관련성을 비교하는 것은 적절하지 않다.

이러한 공분산의 문제점을 보완한 것이 피어슨(Pearson)의 적률 상관 계수이이며, 식 (3)과 같이 계산되어 변수들 간의 상관성을 표준화된 지수로 나타낸다.

$$r_1 = \frac{\sum (x_1 - \bar{x}_1)(y - \bar{y})}{\sqrt{\sum x_1^2}\sqrt{\sum y^2}}$$
$$r_2 = \frac{\sum (x_2 - \bar{x}_2)(y - \bar{y})}{\sqrt{\sum x_2^2}\sqrt{\sum y^2}} \quad (3)$$

여기서 x_1은 잘못된 단위 행동의 수, x_2는 단위 행동 시간, y는 사후 평가 점수를 나타내며, r_1은 사후 평가 점수와 잘못된 단위 행동의 수 간의 상관 계수, r_2는 사후 평가 점수와 단위 행동 시간 간의 상관 계수이다. 상관 계수 값은 1에 가까울수록 두 변수가 양적 선형 관계를 이루고, -1에 가까울수록 음적 선형 관계를 이룬다. 값이 0에 가까우면 두 변수 사이에는 관련성이 없다. 본 실험에서 상관 계수 r_1과 r_2는 표 6과 같이 측정되었으며, 사후 평가 점수가 단위 행동 시간보다 잘못된 단위 행동의 수와 관련이 높다는 결론을 얻을 수 있다. 즉, 기존의 방법 중 수행 시간을 측정하는 방법은 측정하기도 어려울 뿐만 아니라 제안한 방법에 비해 더 좋은 결과를 얻기도 어렵다는 것을 알 수 있다.

Table 6. Comparison of Correlation Coefficients

Task	1	2	3	4	5	6
r_1	0.9599	0.7999	0.7592	0.9921	0.9239	0.9431
r_2	0.7179	0.4981	0.1133	0.8665	0.7873	0.5589

Task	7	8	9	10	11	12
r_1	0.8964	0.8725	0.9947	0.8303	0.9739	0.8903
r_2	0.2116	0.3191	0.1314	0.1739	0.8395	0.1575

4.4 상관 계수의 유의성 검증

상관 분석을 통해 사용자의 사후 평가 점수는 단위 행동의 시간보다 잘못된 단위 행동의 수와 관련이 높다는 것을 알 수 있었다. 하지만, 이 결과가 신뢰성 있는 결과인지를 알아보기 위해 상관 계수의 유의성을 검증하는 과정이 필요하다[13]. 상관 계수의 유의성을 검증하는 방법에는 피셔의 z테스트(Fisher's z-test)와 스테이거의 z테스트

(Steiger's z-test)가 있다. 피셔의 z테스트는 집단 간의 상관 계수의 차이를 검증할 때 사용되는 방법이고[14], 스테이거의 z테스트는 연관성 있는 상관 계수의 차이 검증에 사용된다[15]. 본 실험에서는 동일 집단에 대해 상관 계수를 비교하는 것이므로 스테이거의 z테스트 방법을 사용하였다. 상관 계수의 유의성 검증에서는 자료가 정규 분포를 따른다고 가정하는데, 피어슨의 적률 상관 계수 값은 -1보다 크고 1보다 작은 값이기 때문에 식 (4)와 같이 피셔 변환(Fisher's Transform)을 적용하여 -∞에서 ∞의 값을 가지도록 하였다.

$$z_1 = \frac{1}{2}\ln\left(\frac{1+r_1}{1-r_1}\right)$$
$$z_2 = \frac{1}{2}\ln\left(\frac{1+r_2}{1-r_2}\right) \quad (4)$$

여기서 z_1은 사후 평가 점수와 잘못된 단위 행동의 수 간의 상관 계수를 정규화한 값, z_2는 사후 평가 점수와 단위 행동 시간 간의 상관 계수를 정규화한 값이다.

정규화 된 값을 식 (5)와 같이 Z 점수로 변환하여 상관 계수의 신뢰도를 계산한다. Z 점수의 크기가 클수록 통계적으로 유의미한 차이가 있다고 판단한다.

$$Z = \frac{\sqrt{N-3}(z_1 - z_2)}{\sqrt{2 - 2\frac{Cov(r_1, r_2)}{N(1-r_1^2)(1-r_2^2)}}} \quad (5)$$

표 7은 식 (5)에 의해 계산된 결과를 나타낸다. Z 값이 최소 1.96보다 크기 때문에 단측 검증의 신뢰 수준 p 값은 0.95가 되며, 95% 신뢰 수준으로 사용자의 사후 평가 점수는 단위 행동의 시간보다 잘못된 단위 행동의 수와 관련이 높다고 할 수 있다.

Table 7. Verification of Statistical Significance

Task	1	2	3	4	5	6
Z	4.8691	1.9777	5.2875	8.5522	2.5383	5.0283
p	5.61E-7	2.40E-2	6.20E-8	9.06E-14	5.57E-3	2.47E-7

Task	7	8	9	10	11	12
Z	5.1927	4.1380	10.5877	4.2779	4.5802	5.1854
p	1.04E-7	1.75E-5	1.01E-25	9.43E-6	2.32E-6	1.08E-7

5. 결 론

사용성 평가 방법은 다양하게 존재하지만, 평가 방법에 맞추어 전문적인 지식과 특정 장비가 필요하거나 많은 시간 동안 사용자가 함께 있어야 하는 문제점이 있다. 이를 해결하여 시간과 비용, 노력을 줄이기 위한 연구가 진행되어 왔지만, 데이터가 오염될 수 있거나, 작업의 성공을 객관적으로 판별할 수 없거나, 웹 페이지에만 적용할 수 있는 등의 한계가 있다.

이러한 기존 연구의 한계를 보완하고 사용자 평가 시 문제가 되는 작업 및 사용자를 선별하기 위해, 사용자의 행동을 측정하는 단위 행동을 정의하고 객체 기반의 애플리케이션에 적용할 수 있는 손실률을 제안하였다. 기존의 방법과 제안한 방법을 비교하기 위하여 문제 없이 작업을 수행한 사용자의 데이터를 기반으로 기존의 손실률을 계산하였을 때 높은 손실률이 발생함을 보임으로써 기존 방법의 문제점을 지적하였다. 이에 반해 제안한 손실률은 0으로 계산되어 사용자가 수행한 작업에 문제가 없음을 잘 나타내고 있음을 보였다. 또한, 상관 분석을 통해 사용자의 사후 평가 점수가 수행 시간보다 잘못된 행동의 수와 관련이 높다는 것을 보임으로써 제안한 단위 행동 기반의 손실률이 타당함을 보였다.

사용성 평가는 대상이 되는 애플리케이션의 사용성 자체를 평가하는 데에도 목적이 있지만, 문제가 되는 부분을 발견하여 애플리케이션을 개선하기 위한 목적도 있다. 문제가 되는 부분을 발견하기 위한 일반적인 방법은 녹화된 영상 또는 직접 관찰을 통해 사용자가 수행에 어려움을 겪는 작업을 찾는 것이다. 하지만 모든 사용자 및 모든 작업에 대한 영상을 분석하는 것은 많은 시간과 노력이 필요하기 때문에 비용이 발생한다. 사용성 평가를 위해 수행할 작업과 단위 행동을 정의하고, 제안한 손실률을 계산하면 문제가 되는 단위 행동을 쉽게 선별할 수 있다. 이렇게 선별된 단위 행동에 대한 녹화 영상만 분석하면 되기 때

문에 시간과 노력, 비용을 절약할 수 있다.

본 논문에서는 PC 또는 모바일 환경에서 스크래치, 스택 미니 등과 같은 객체 기반의 애플리케이션에 대한 사용성 평가 방법을 제안하였는데, 문제가 되는 단위 행동을 쉽게 선별할 수 있다는 장점은 있지만, 선별된 단위 행동이 어떠한 원인에 의한 것인지를 알기 위해서는 영상 분석 등 부가적인 분석 방법에 대한 연구가 필요하다. 또한 스크립트 또는 고급 프로그래밍 언어 등 텍스트 기반의 코딩 교육에서 사용하는 통합개발환경의 경우 메뉴, 툴바 등의 요소에 대해서는 제안한 방법으로 사용성 평가가 가능하지만 입력된 텍스트 명령에 대한 평가는 가능하지 않기 때문에 텍스트를 분석하여 단위 행동을 추출하고 평가하는 방법에 대한 연구가 필요하다.

References

[1] P.W. Jordan, Designing Pleasurable Products, Taylor & Francis, London and New York, 2000, pp. 1-10.

[2] S.-H. Cho, "The effect of robots in education based on STEAM," Journal of Korea Robotics Society, vol. 8, no. 1, pp. 58-65, March 2013.

[3] Y.A. Kim, K.H. Chae, Y.-J. Sohn, J.-M. Yang, and C.D. Koo, "Teachers and students' recognition about learning with a humanoid robot in elementary school," Journal of Korea Robotics Society, vol. 9, no. 3, pp. 185-195, September 2014.

[4] J. Nielsen, Usability Engineering, Morgan Kaufmann, 1933, pp. 24-40.

[5] A. Holzinger, "Usability engineering methods for software developers," Communications of the ACM, vol. 48, no. 1, pp. 71-74, January 2005.

[6] J. Nielsen, "Finding usability problems through heurisitc evaluation," in CHI '92 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1982, pp. 373-380.

[7] J. Hom, The Usability Methods Toolbox Handbook, 1988, pp. 21-44, http://jthom.best.vwh.net/usability/usable.htm

[8] T. Tullis and B. Alvert, Measuring the User Experience, Morgan Kaufmann, 2009, pp. 97-106.

[9] DataLogger webpage, http://www.userfocus.co.uk/resources/datalogger.html

[10] P.A. Smith, "Toward a practical measure of hypertext usability," Interacting with Computers, vol 8, no. 4, pp. 365-381, December 1996.

[11] M. Resnick, J. Maloney, A. Monroy-Hernandez, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: programming for All," Communications of the ACM, vol. 52, no. 11, pp. 60-67, November 2009.

[12] C. Villamor, D. Willis, and L. Wroblewski, Touch gesture reference guide webpage, http://www.lukew.com/touch

[13] D.A. Kenny, Statistics for Social and Behavioral Sciences, Little, Brown, 1987, pp. 270-288.

[14] J.H. Steiger, "Tests for comparing elements of a correlation matrix," Psychological Bulletin, vol. 87, no. 2, pp. 245-251, March 1980.

[15] J.F. Ehlers, Cybernetic Analysis for Stocks and Futures: Cutting-Edge DSP Technology to Improve Your Trading, Wiley, 2004, pp. 1-10.

박 희 성

2013 광운대학교 정보제어공학과 (공학사)

2015 광운대학교 제어계측공학과 (공학석사)

2015~현재 삼성전자 S/W 센터 차세대 Interaction 팀

관심분야 : UX, HRI

박 광 현

1994 KAIST 전기및전자공학과 (공학사)

1997 KAIST 전기및전자공학과 (공학석사)

2001 KAIST 전기및전자공학과 (공학박사)

2008~현재 광운대학교 로봇학부 부교수

관심분야 : 교육용 로봇, HRI, 로봇 소프트웨어