

HCI 장치의 사용자 인터페이스 규칙스키마에 관한 연구

김흥규*

A Study on Rule Schemas of User Interface in HCI Devices

Heung-Kyu Kim*

요 약

인간의 많은 작업이 컴퓨터를 이용하여 수행됨에 따라 HCI(Human-Computer Interaction)에 대한 연구가 활발히 진행되고 있다. HCI 연구의 한 가지 목표는 좀 더 일관성 있는 시스템의 설계를 제안하는 것이다. 사용자의 관점에서 일관성을 평가하기 위한 대표적인 인지모형으로 TAG(Task Action Grammar), GOMS(Goals - Operators - Methods - Selection Rule), 그리고 GTN(General Transition Network)이 있다. 이 중 TAG는 규칙스키마의 개수를 가지고 시스템의 일관성을 평가한다. 즉, 사용자가 시스템을 이용하여 작업을 할 때 익혀야 할 규칙스키마의 개수가 적으면 적을수록 그 시스템의 일관성은 높다는 것이다. 이 논문에서는 TAG에서 시스템의 일관성을 측정할 때 단순히 규칙스키마의 개수뿐만 아니라 규칙스키마 사이의 거리도 역시 중요한 고려 대상이 되어야 한다는 것을 보이고자 한다. 즉, 규칙스키마 사이의 거리가 짧으면 짧을수록 모든 규칙스키마를 습득하기가 쉬워질 것이다. 실험결과 이 가설은 확인되었다. 따라서 시스템을 설계할 때 규칙스키마의 개수뿐만 아니라 규칙스키마 사이의 거리도 고려되어야 할 것이다.

▶ Keywords : HCI, 사용자 인터페이스, TAG, 규칙스키마

Abstract

As humans perform many tasks using computers, HCI(Human-Computer Interaction) has received much attention these days. One objective of HCI research is to propose how to design more consistent systems. In order to evaluate the consistency of HCI devices from the user's perspective, several models such as TAG(Task Action Grammar), GOMS(Goals-Operators-Methods-Selection Rule), and GTN(General Transition Network) have been developed. TAG specifies actions to perform tasks in terms of rule schema. It has been verified that the less the number of rule schema is, the better users perform the tasks due to

• 제1저자 : 김흥규

• 투고일 : 2012. 9. 20, 심사일 : 2012. 10. 30, 게재확정일 : 2012. 11. 21.

* 단국대학교 경영학부(School of Business Administration, Dankook University)

※이 연구는 2012년도 단국대학교 대학연구비 지원으로 연구되었음.

assumably higher consistency. This paper hypothesizes that the consistency of systems depends not only on the number of rule schema but also on the distances between rule schema. That is, the closer the rule schema are, the easier it is to acquire the whole set of rule schema. An experiment supported this hypothesis. Therefore, distances between rule schema should be considered as well as the number of rule schema when designing systems.

▶ Keywords : HCI, User Interface, TAG, Rule Schema

I. 서 론

HCI(Human-Computer Interaction)란 인간과 컴퓨터가 쉽고 편하게 상호작용할 수 있도록 인터페이스를 디자인하고 평가하는 학문으로서 이 과정을 둘러싼 중요 현상들에 관한 연구도 포함한다.

최근 HCI에 대한 연구가 활발하게 진행되고 있고, 인간의 거의 모든 작업이 컴퓨터를 이용하여 수행되고 있는 현실을 감안하면 HCI연구의 중요성은 날로 더해 갈 것이다 [2].

인간이 서로 다른 인터페이스를 가진 여러 시스템을 동시에 사용한다면 이러한 인터페이스 사이의 비일관성으로 인하여 혼란과 오류를 초래하게 되고, 그 결과 생산성이 떨어질 것이다 [1]. 이러한 현상은 인간의 작업환경에서 뿐만이 아니고 작업 이외의 환경에서도 일어나고 있다[3]. 따라서 데스크톱, 모바일 기기, 가상공간 등은 비록 서로 다른 플랫폼에서 개발되더라도 일관된 인터페이스를 갖도록 설계하고 구현할 필요가 있다.

이 논문의 구성은 다음과 같다. 제2절에서는 시스템의 일관성을 측정하기 위한 모형인 TAG, GOMS, 그리고 GTN에 관하여 개략적으로 살펴본다. 제3절에서는 TAG모형과 TAG 모형에서 규칙스키마 사이의 거리를 표현하는 측정지표에 관하여 살펴본다. 제4절에서는 규칙스키마 사이의 거리가 서로 다른 세 개의 시스템을 고안한다. 제5절에서는 제4절에서 고안한 세 개의 시스템을 이용하여 실험을 수행하고 이의 결과를 분석하여 이 논문에서 보이고자 하는 가설을 검증한다. 마지막으로 제6절에서는 제5절의 내용으로부터 어떠한 시사점을 얻을 수 있는가를 살펴본다.

II. 기존 연구

HCI연구에서 근간이 되는 학문분야는 인지공학인데 이는

인간의 정보처리과정을 연구하여 이를 시스템의 설계에 응용하여 시스템을 통한 생산성을 극대화하고자하는데 그 목적을 두고 있다 [4], [5], [8]. 인지공학은 HCI연구에서 한 가지 기본원리를 제공했는데 이는 다음과 같다.

인간의 정보처리과정에서의 가장 큰 약점은 작업기억장소의 한계이다. 즉 인간의 작업기억장소는 용량에 제한이 있어 기억항목의 수가 증가함에 따라 지수적으로 인지적 부하가 증가하게 된다는 것이다. 따라서 시스템의 설계 시 작업기억장소에 걸리는 인지적 부하를 줄여주는 방안을 우선적으로 고려해야 한다 [5], [6].

인지공학은 작업기억장소에 걸리는 인지적 부하를 줄여주는 방안의 하나로 일관성 있는 시스템의 설계를 제안하게 되었다 [5]. 이러한 설계지침을 따르기 위해서는 먼저 시스템의 일관성을 측정할 수 있어야 하는데 이러한 목적으로 개발된 몇 가지 대표적인 모형으로는 TAG(Task Action Grammar), GOMS(Goals-Operators-Methods-Selection rules), 그리고 GTN (General Transition Network)이 있다.

TAG는 사용자가 갖고 있는 시스템의 개념적인 모델로부터 그 시스템을 사용하여 원하는 작업을 수행하기 위한 행동 과정을 설명하기 위한 모형이다. TAG는 수행하여야 할 작업으로부터 이에 필요한 행동순서를 얻게 되는 과정을 규칙스키마를 통하여 표현한다. 여기서 구한 규칙스키마의 개수로 그 시스템의 일관성을 평가한다 [11].

GOMS는 사용자가 시스템으로부터 이루고자 하는 작업을 수행하기 위해 필요한 일련의 방법을 나타내고 각 방법은 사용자가 행하여야 할 작동으로 구성된다. 여기서 방법이 둘 이상이면 상황에 맞는 방법을 선택하는 과정을 설명하는 선택규칙을 포함한다 [9].

마지막으로 GTN은 시스템의 상태와 그 상태에서 가능한 조건과 행동을 도표로 나타낸다 [10].

한편, GOMS는 분석자의 주관이 크게 작용하여 객관성을 잃는 단점이 있고, GTN은 분석하고자 하는 시스템이 조금이라도 복잡해지면 표현하기가 어려운 단점이 있는데 반해,

TAG는 여러 작업을 그 작업을 이루는 행동의 유사성을 가지고 몇 가지의 규칙스키마로 유도하여 단순화시키므로 실용적이라는 장점이 있다.

현재 TAG는 규칙스키마의 개수를 가지고 시스템의 일관성을 평가한다. 즉, 사용자가 시스템을 이용하여 작업을 할 때 익혀야 할 규칙스키마의 개수가 많으면 많을수록 그 시스템의 일관성은 낮아진다는 것이다. 그러나 규칙스키마의 개수만으로 시스템의 일관성을 평가하기에는 부족한 면이 있다.

이 논문에서는 TAG에서 시스템의 일관성을 측정할 때 단순히 규칙스키마의 개수뿐만 아니라 규칙스키마 사이의 거리도 역시 중요한 고려대상이 되어야 한다는 것을 보이고자 한다.

III. TAG

3.1 TAG의 개요

TAG는 사용자의 입장에서 각 작업과 그 작업을 수행하기 위해 필요한 행동 사이의 관계를 정형화된 방식으로 표현하는 모형이다. 각 작업은 특성치의 조합으로 표현되어 진다. 예를 들어 매킨토시의 그림그리기 소프트웨어인 MacDraw에서 대표적인 특성치와 정사각형을 그리는 작업은 다음과 같이 표현된다.

Feature set	
Effect:	add, remove, move
Type of object:	rectangle, ellipse, line, freehand-shape
Constraint:	yes, no
Draw a square:	
Effect	= add
Type of object	= rectangle
Constraint	= yes

위의 작업을 수행하기 위해 필요한 행동은 다음과 같은 작업 규칙으로 표현된다.

Draw a square:
task[Effect = add, Type of object = rectangle, Constraint = yes] := select-tool[Type of object=rectangle] + press SHIFT + place mouse ...
select-tool[Type of object=rectangle] := '사각형 툴'을 선택한다.

위의 규칙은 적당한 그림 툴을 선택하는 하부 작업 규칙인 select-tool, SHIFT키를 누르는 행동, 그리고 그리고자 하는 정사각형의 위치와 크기를 정하는 행동으로 이루어져 있다.

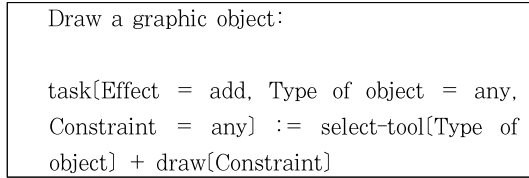
직사각형을 그리는 작업도 위와 유사한 방법으로 다음과 같은 작업 규칙으로 나타낼 수 있다.

Draw a rectangle:
task[Effect = add, Type of object = rectangle, Constraint = no] := select-tool[Type of object=rectangle] + place mouse ...

위의 두 가지 작업 규칙은 draw라는 하위 작업 규칙을 도입하여 하나의 작업 규칙으로 다음과 같이 일반화할 수 있다.

Draw a rectangle or square:
task[Effect = add, Type of object = rectangle, Constraint = any] := select-tool[Type of object=rectangle] + draw[Constraint]
draw[Constraint = yes] := press SHIFT + place mouse ...
draw[Constraint = no] := place mouse ...

위의 작업 규칙은 그림 툴을 이용한 모든 작업을 포함하도록 다음과 같은 작업 규칙으로 한 단계 더 일반화할 수 있다. 이는 주어진 과제의 특성으로부터 취해야 할 행동을 도출하기 위한 규칙스키마의 최상위 수준이 된다.



위와 같이 TAG는 사용자의 작업 지식을 정형화된 방식으로 나타낸 후 각 작업을 수행하기 위해 필요한 행동 양식의 동일성에 의해 개별 작업을 집산화하여 각 집단에 속하는 작업을 대표하는 하나의 규칙스키마를 유도해내는 분석과정을 거친다. 따라서 규칙스키마의 수가 적을수록 시스템의 일관성은 높아진다. 이와 같이 TAG는 사용자의 기억특성을 간단하게 나타내는 장점을 갖고 있다.

한편, TAG는 몇 가지 단점도 갖고 있다. 첫째로 내부 인지적 동작을 나타내지 못한다. 따라서 내부 인지적 동작을 많이 필요로 하는 시스템의 일관성을 나타내는 데에는 부적합하다. 다음으로 규칙스키마의 개수만으로는 시스템의 일관성을 잘 나타낼 수 없다. 이는 규칙스키마의 개수가 같더라도 규칙스키마 사이의 거리가 짧은 시스템일수록 사용자는 그 시스템의 일관성이 더 높다고 느낄 것이기 때문이다. 이는 이 논문에서 확인해보고자 하는 가설이다.

3.2 규칙스키마 사이의 거리

TAG에서의 규칙스키마는 하부 규칙스키마나 행동을 노드로 갖는 연결리스트로 나타낼 수 있다. 예를 들어 MacDraw에서 그림 툴을 가지고 행하는 작업을 나타내는 규칙스키마는 다음과 같은 연결리스트로 나타낼 수 있다.

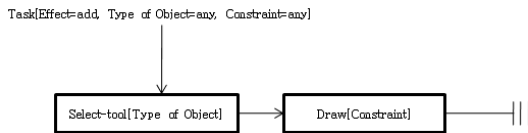


그림 1. 연결리스트로 나타낸 규칙스키마
Fig 1. Linked List Representing Rule Schema

두 연결리스트의 노드 사이에 거리가 있는 경우는 다음의 네 가지이다.

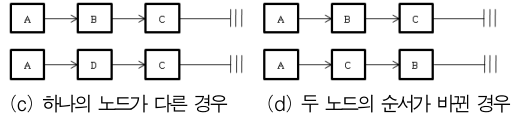
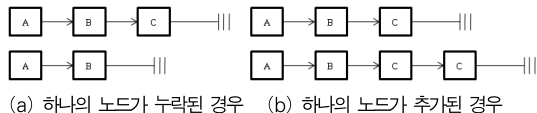


그림 2. 두 연결리스트의 노드 사이에 거리가 있는 경우
Fig 2. Types of Distances between Nodes in Linked Lists

위 네 가지 항목의 연속된 조작으로 임의의 연결리스트를 임의의 다른 연결리스트로 바꿀 수 있다. 따라서 이러한 과정에서 요구되는 항목의 최소 개수를 두 연결리스트 사이의 거리를 나타내는 지표로 이용할 수 있다.

두 연결리스트 사이의 거리를 위와 같이 정의할 때 이는 식(1)과 같은 동적계획법 문제를 풀어 구할 수 있다. 식(1)에서 $f(i, j)$ 는 노드 $s_1, s_2, s_3, \dots, s_i$ 의 순으로 되어 있는 연결리스트와 노드 $t_1, t_2, t_3, \dots, t_j$ 의 순으로 되어 있는 연결리스트 사이의 거리를 나타내는 함수이다.

$$\begin{aligned}
 f(0, 0) &= 0 && \dots \text{식(1)} \\
 f(i, j) &= \min [f(i-1, j) + 1, \\
 & f(i, j-1) + 1, \\
 & f(i-1, j-1) + d(s_i, t_j), \\
 & f(i-2, j-2) + d(s_{i-1}, t_j) + d(s_i, t_{j-1})]
 \end{aligned}$$

where $d(s_i, t_j) = 0$ if $s_i = t_j$
 $= 1$ otherwise

두 연결리스트 사이의 거리를 구하는 방법을 이용하여 TAG를 통하여 얻은 규칙스키마 $T_1, T_2, T_3, \dots, T_n$ 전체의 거리를 다음과 같이 구할 수 있다.

먼저 두 규칙스키마 사이의 거리를 구하여 다음의 행렬을 구할 수 있다. 여기서 대각선에 해당되는 항은 해당 규칙스키마 자기 자신과의 거리를 나타내므로 0이 된다.

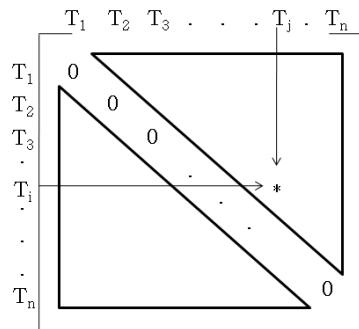


그림 3. 규칙스키마 사이의 거리를 나타내는 행렬
Fig 3. Matrix of Distances between Rule Schemas

[그림 3]에서 *는 규칙스키마 T_i 와 T_j 사이의 거리를 나타낸다. 규칙스키마 T_i 와 T_j 사이의 거리는 규칙스키마 T_j 와 T_i 사이의 거리와 같기 때문에 위의 행렬은 대칭 행렬이 된다. 따라서 위의 행렬 중 임의의 한 쪽 삼각형 안에 있는 항만 고려하면 된다. 이 삼각형 안에는 $nC2$ 개의 항이 존재하므로 삼각형 안의 모든 항을 더하여 $nC2$ 로 나누어 주면 전체 규칙스키마 사이의 평균 거리가 되며, 이는 전체 규칙스키마 사이의 거리가 시스템의 일관성에 미치는 영향을 나타낸다. 그리고 시스템에서 임의의 한 규칙스키마와 그 외 규칙스키마 사이의 평균 거리는 이 규칙스키마에 해당하는 행이나 열의 모든 항을 더하여 $(n-1)$ 로 나누어 구할 수 있다. 이는 이 규칙스키마가 다른 규칙스키마와의 거리에 의해 시스템의 일관성에 미치는 영향을 나타낸다.

IV. 가상 시스템의 설계

피실험자가 실험에 흥미를 갖고 참여하도록, 그리고 사용 경험이 전혀 없어서 같은 교육으로 같은 정도의 정보를 가질 수 있도록 하기 위하여 가상적인 우주선 안에서의 조종간을 설계하여 실험에 이용한다. 이 조종간은 기능별로 크게 3가지로 나눌 수 있고 이는 다음과 같다.

1. 운전기능 (동력 전달장치, 속도 조절장치, 방향 제어장치)
2. 디스플레이제어기능 (디스플레이 모드 선정 장치, 화면조정장치)
3. 무기 이용 기능 (목표물 선정 장치, 무기 선택 장치, 발사장치)

이 시스템을 사용하여 행하는 작업을 작업방식에 따라 분류해보면 다음과 같다.

1. 한 번의 키 입력으로 끝나는 작업
one key → one level
one (toggle) key → two level
2. 한 키를 어느 정도 계속 눌러서 수행하는 작업
one duration (continuous) key → all level
3. 몇 단계의 키 조작으로 수행하는 작업
4. 몇 단계의 키 조작과 조이스틱의 조합으로 수행하는 작업

이 시스템에서 작업이 복잡해지면 위의 1번과 2번의 작업 방식으로는 불가능하고 몇 단계의 키 조작이 필수적이므로 본 연구에서는 3번과 4번의 작업방식으로 행해지는 수행 작업을 분석 대상으로 한다.

시스템 I에서는 3번과 4번의 작업방식에 해당하는 규칙스키마 사이의 거리가 가능한 한 최소치를 갖도록 설계하였다. 그리고 시스템 II와 시스템 III에서는 규칙스키마 사이의 거리가 의미 있는 수준 안에서 좀 더 크도록 작업방식을 변경하였다.

4.1 시스템 I

시스템 I에서 수행 가능한 작업은 다음과 같은 유형으로 나눌 수 있다.

1. 몇 가지 수준이나 형태 중에서 한 가지를 선택하는 작업
2. 연속적인 변화치를 조절하는 작업
3. 화면에서 원하는 부분을 선택하는 작업

시스템 I에서 수행되는 작업은 다음과 같은 규칙스키마로 분류할 수 있다.

Rule Schemas I: task rules

T1: = "key"

T2: = "key(toggle)"

T3: = "key(%duration)"

T4: = "key" + number-select + "select"

T5: = "key" + joystick-menu(object, value) + "select"

T6: = "key" + joystick-position(object, effect, position) + "select"

위의 TAG 표현에서 규칙스키마 T4에서 T6까지로 나타나는 작업이 이 실험에서 고려하고자 하는 규칙스키마이다. 시스템 I에서 전체 규칙스키마 사이의 평균 거리는 2가 되고 규칙스키마 T4에서 T6 사이의 평균 거리는 두 규칙스키마 사이의 거리가 가질 수 있는 수준 중에서 최소치인 1이 된다.

4.2 시스템 II

어떠한 작업이 시스템 I의 규칙스키마 T4에서 T6까지로

나타나는지를 살펴보면 다음과 같다.

T4: engine booster, engine #, size
 T5: speed, weapon
 T6: zoom, target

시스템 II에서는 규칙스키마 T4에서 T6까지를 다음과 같이 수정하였다.

첫 번째로 T4의 규칙스키마로 나타나는 작업의 유형은 먼저 해당키를 입력하여 원하는 모드로 들어가서 화면에 나타나는 메뉴 중에서 원하는 항목에 해당되는 숫자 키를 입력하고 선택키로 확인과 더불어 현재의 모드를 빠져 나오는 형태이다. 이 규칙스키마는 시스템 I과 같도록 그대로 둔다.

두 번째로 T5의 규칙스키마로 나타나는 작업의 유형은 먼저 해당키를 입력하여 원하는 모드로 들어가서 화면에 나타나는 메뉴 중에서 조이스틱 중 왼쪽 방향키와 오른쪽 방향키를 조절하여 반전된 지시자를 원하는 항목으로 옮겨 선택하고 선택키로 확인과 더불어 현재의 모드를 빠져 나오는 형태이다. 이 규칙스키마도 시스템 I과 같도록 그대로 둔다.

세 번째로 T6의 규칙 스키마로 나타나는 작업의 유형은 먼저 해당키를 입력하여 원하는 모드로 들어가서 화면에 나타나는 표시자를 조이스틱을 이용하여 원하는 곳으로 이동시킨 후 선택 버튼을 눌러 확인과 더불어 현재의 모드를 빠져 나오는 형태이다. 이 규칙스키마는 다음과 같은 형태로 수정한다.

T6' := "key" + joystick-position(object, effect, position) + "position"

즉, 시스템 I에서는 선택키를 누름으로써 조이스틱으로 선택한 위치의 확인과 더불어 현재의 모드를 빠져 나오는데 반해 시스템 II에서는 위치키를 누르도록 차별화시켰다.

4.3 시스템 III

시스템 III에서는 가능한 한 최대의 차별성 수준을 갖도록 다음과 같이 수정하였다.

첫 번째로 T4의 규칙 스키마는 다음과 같이 수정한다.

T4'' := "key" + number-select

즉, 시스템 III에서는 선택키가 필요 없이 숫자 키로 해당되는 항목을 선택하는 것으로 작업을 끝내는 형태로 원래의

작업절차를 단순화하였다.

두 번째로 T5의 규칙 스키마는 다음과 같이 수정한다.

T5'' := "key" + joystick-menu(object, value) + "select" + "reset"

즉, 시스템 I에서는 선택키를 누름으로써 확인과 더불어 현재의 모드를 빠져 나오는데 반해 시스템 III에서는 선택키를 눌러 확인을 하고 다시 빠져 나오기 버튼을 누름으로써 현재의 모드를 벗어나도록 한다.

세 번째로 T6의 규칙스키마는 시스템 II와 같은 형태로 수정한다.

4.4 세 가지 시스템의 비교

지금까지 논의한 세 시스템에서 전체 규칙스키마 사이의 평균 거리와 규칙스키마 T4부터 T6까지의 평균 거리를 계산한 결과는 <표 1>에 나타낸 바와 같다.

표 1. 세 가지 시스템의 비교
 Table 1. Comparisons among Three Systems

	시스템 I	시스템 II	시스템 III
규칙 스키마	T1 := "key"	T1' := "key"	T1'' := "key"
	T2 := "key(toggle)"	T2' := "key(toggle)"	T2'' := "key(toggle)"
	T3 := "key(%duration)"	T3' := "key(%duration)"	T3'' := "key(%duration)"
	T4 := "key" + number-select + "select"	T4' := "key" + number-select + "select"	T4'' := "key" + number-select
	T5 := "key" + joystick-menu(object, value) + "select"	T5' := "key" + joystick-menu(object, value) + "select"	T5'' := "key" + joystick-menu(object, value) + "select" + "reset"
	T6 := "key" + joystick-position(object, effect, position) + "select"	T6' := "key" + joystick-position(object, effect, position) + "position"	T6'' := "key" + joystick-position(object, effect, position) + "position"
T1~T6	2	2.1	2.3
T4~T6	1	1.7	2.7

V. 실험

전술한 바와 같이 TAG에서 시스템의 일관성을 평가할 때

단지 규칙스키마의 개수만이 아니고 규칙스키마 사이의 거리도 고려하여야 한다. 즉, 규칙스키마 사이의 거리가 짧으면 짧을수록 전에 익힌 규칙스키마로 인한 정의 학습효과로 인해 그 다음 규칙스키마를 익히기가 쉽고 일정 시간이 흐른 후 그 시스템을 다시 사용할 때 숙지한 규칙스키마로부터 잊어버린 규칙스키마를 쉽게 유추할 수 있으므로 사용자는 그 시스템의 일관성이 높다고 느낄 것이다. 한편, 규칙스키마 사이의 거리가 길면 길수록 정의 학습효과를 기대할 수 없을 뿐만 아니라 일정 시간이 흐른 후 그 시스템을 다시 사용할 때 이미 알고 있는 규칙스키마로부터 잊어버린 규칙스키마를 유추하기가 어려우므로 그 시스템은 일관성이 낮다고 느낄 것이다.

이러한 가설 하에 본 실험에서는 제3절에서 고안한 세 가지 시스템을 이용한다.

5.1 실험설계

먼저 매킨토시용 응용소프트웨어인 Hyper Card를 이용해 세 가지 시스템을 나타내는 시뮬레이터를 구현하였다. 시스템 I을 나타내는 시뮬레이터에서 목표물을 선정하는 작업에 대한 디스플레이와 키 조작 과정은 [그림 4]에 나타낸 바와 같다.



그림 4. 시스템 I을 나타내는 시뮬레이터
Fig 4. Simulator for System I

여기서는 매킨토시를 사용해 본 경험이 없는 20대 중반의 경영학전공 대학원 학생 15명을 세 집단으로 나누어 실험하였다. 첫 번째 집단은 시스템 I의 시뮬레이터를 먼저 사용한 후 일주일 간격으로 시스템 II, III의 시뮬레이터를 사용하도록 하였고, 두 번째 집단은 시스템 II의 시뮬레이터를 먼저 사용한 후 일주일 간격으로 시스템 III, I의 시뮬레이터를 사용하도록 하였고, 그리고 세 번째 집단은 시스템 III의 시뮬레이터를 먼저 사용한 후 일주일 간격으로 시스템 I, II의 시뮬레이터를 사용하도록 하였다. 이는 피실험자 사이의 개인차로 인해 우리가 원하는 실험결과와 왜곡을 방지하기 위한 것이다.

피실험자는 첫 번째 실험을 위한 교육에서 시뮬레이터의 작동방법을 설명하는 매뉴얼과 연습시뮬레이터를 이용한 방법으로 교육하였고, 두 번째 실험을 위한 교육에서는 시뮬레이터의 그림과 매뉴얼을 이용하여 교육하였고, 그리고 세 번째 실험을 위한 교육에서는 단지 매뉴얼만을 이용하여 먼저 접한 시스템의 전이효과를 가능한 한 줄이도록 노력하였다. 교육 후에 시행한 시험에서는 컴퓨터 로그를 이용해 각 문제를 풀면서 누른 키를 기록하도록 하였다.

표 2. 교육, 실험시간, 그리고 실험순서
Table 2. Training, Experiment Time, and Experiment Sequence

	교육	실험 #1	교육	실험 #2	교육	실험 #3
시간	20분	10분	10분	10분	5분	10분
집단1		시스템 I		시스템 II		시스템 III
집단2		시스템 II		시스템 III		시스템 I
집단3		시스템 III		시스템 I		시스템 II

실험에서는 10개의 문제를 사용하였고, 이 중에서 6개가 분석하고자 하는 문제이다. 나머지는 문제가 간단하여 단순한 키를 선택하는 것으로 끝나는 것이고 모든 시스템에서 작동방식이 똑같은 작업이다. 분석하고자 하는 6개의 문제 중에서 두 개는 T4, T4', T'의 규칙스키마로 나타나는 작업이다. 또 다른 두 개는 T5, T5', T5''의 규칙스키마로 나타나는 작업이다. 그리고 마지막 두 개는 T6, T6', T6''의 규칙스키마로 나타나는 작업이다.

5.2 실험결과

피실험자가 각 문제를 풀면서 누른 키의 기록을 가지고 오류의 종류와 개수를 계산하여 이를 통계패키지 SPSS

v12.0k를 이용하여 분석하였다. 한편, 오류의 종류는 크게 각 문제를 푸는데 필요한 키의 입력이 빠지거나 불필요한 키의 입력이 추가된 두 가지의 경우로 나눌 수 있다.

실험의 결과, 시스템 I의 경우 총 오류의 수는 5개였고, 시스템 II의 경우 총 오류의 수는 16개였다. 그리고 시스템 III의 경우는 총 오류의 수가 42개나 되었다. 먼저 시스템에 따른 오류수의 다중범위분석을 한 결과는 <표 3>에 나타난 바와 같다.

표 3. 시스템 간의 오류수의 다중범위분석
Table 3. Multiple Range Analysis on Error # among Systems

수준	개수	평균	표준오차	신뢰구간 (95%)
시스템 I	15	0.3	0.2	(0.0~0.8)
시스템 II	15	1.1	0.2	(0.6~1.5)
시스템 III	15	2.8	0.3	(2.3~3.2)

시스템 I의 평균 오류수는 단지 0.3개뿐인데 반해 시스템 II의 평균 오류수는 1.1개이고 시스템 III의 경우는 평균 오류수가 2.8개나 되어 시스템에 따라 큰 차이를 나타내고 있다. 이는 시스템에 따른 차이가 크다는 것을 암시한다.

다음으로 집단에 따른 오류수의 다중범위분석을 한 결과는 <표 4>에 나타난 바와 같다.

표 4. 집단 간의 오류수의 다중범위분석
Table 4. Multiple Range Analysis on Error # among Groups

수준	개수	평균	표준오차	신뢰구간 (95%)
집단1	15	1.4	0.4	(0.9~1.9)
집단2	15	1.4	0.3	(0.9~1.9)
집단3	15	1.4	0.3	(0.9~1.9)

공교롭게도 집단 1, 2, 3 모두 평균 오류수는 1.4개로 같다. 이는 집단에 따른 차이가 거의 없음을 암시한다.

그리고 결과변수를 피실험자가 범한 오류의 수로 하고 각 집단이 해당 시스템에 노출된 순서를 인자로 하는 반복이 있는 일원배치법으로 생각하여 분산분석을 실시한 결과는 <표 5>에 나타난 바와 같다.

표 5. 시스템의 노출 순서를 인자로 한 분산분석표
Table 5. ANOVA Table with System Exposure Sequence

요인	제곱합	자유도	제곱 평균	F비	유의 확률
노출 순서	0.1333	2	0.067	0.037	0.964

진차	76.667	42	1.825		
계	76.800	44			

분석결과 시스템의 노출순서가 결과변수에 미치는 영향은 거의 없는 것으로 나타났다. 즉, <표 5>는 피실험자에게 각 시스템 사이의 노출간격을 1주일로 하고 두 번째 시스템에 노출시킬 때에는 연습시뮬레이터를 사용하지 않고 시뮬레이터의 그림과 매뉴얼만 가지고 교육시키고, 세 번째 시스템에 노출시킬 때에는 단지 매뉴얼만 가지고 교육시킨 것이 전이효과를 거의 배제하여 각 시스템에 노출될 때 같은 정도로 시스템에 대한 지식을 갖게 하였음을 의미한다.

마지막으로 결과변수를 오류의 수로 하고 시스템의 종류를 하나의 인자로 하고 집단을 다른 인자로 하는 반복이 있는 이원배치법으로 생각하여 분산분석을 실시한 결과는 <표 6>에 나타난 바와 같다.

표 6. 시스템과 집단을 인자로 한 분산분석표
Table 6. ANOVA Table with System and Group

요인	제곱합	자유도	제곱 평균	F비	유의 확률
시스템	48.133	2	24.067	30.943	0.000
집단	0.000	2	0.000	0.000	1.000
교호 작용	0.667	4	0.167	0.214	0.929
진차	28.000	36	0.778		
계	76.800	44			

분석결과 어떠한 유의수준 하에서도 각 시스템 사이에는 유의한 차이가 있는 것으로 나타났다. 이는 이 논문에서 제시한 가설의 타당성을 입증하고 있다. 한편, 집단의 영향이나 시스템과 집단과의 교호작용이 결과변수에 미치는 영향은 거의 없는 것으로 나타났다. 다시 말해서 <표 6>은 단지 시스템에 따른 차이, 즉 규칙스키마 사이의 거리만이 피실험자가 행한 오류의 수에 영향을 준다는 것을 의미한다.

VI. 결론

HCI연구에서 시스템의 일관성을 측정하려는 모형 중의 하나인 TAG는 사용자의 입장에서 작업과 그 작업을 수행하기 위해 필요한 행동 사이의 관계를 정형화된 방식으로 나타낸 다음 각 작업을 수행하기 위해 필요한 행동 양식의 유사성에 의해 개별 작업을 집단화하여 각 집단을 대표하는 하나의 규칙스키마를 도출하는 분석절차를 거친다. 즉, TAG는 인간의 스키마 개념을 이용하여 시스템의 일관성을 측정하려는 모형

이다.

이 논문에서는 시스템의 일관성이 단지 규칙스키마의 개수 뿐만이 아니라 규칙스키마 사이의 거리에도 의존한다는 것을 보여주고자 하였다. 이를 위하여 피실험자가 훈련을 통해 같은 정도의 지식을 갖도록 하기 위하여 피실험자가 접해 보지 못한 가상의 시스템을 규칙스키마 사이의 거리가 최소가 되도록 설계하고 다시 규칙스키마 사이의 거리만이 다르도록 하기 위해 전체 하위 규칙스키마와 행동의 개수가 같고 어느 정도의 부하가 걸리는지 예측하기가 힘든 하위 규칙스키마와 행동은 같게 유지하면서 규칙스키마 사이의 거리가 좀 더 긴 두 개의 시스템을 고안하였다. 이를 이용하여 본 논문에서 제시한 가설을 확인하기 위한 실험을 수행하였다. 실험결과가 보여주듯이 규칙스키마 사이의 거리가 가장 짧은 처음 시스템에서 피실험자가 행한 오류의 수가 훨씬 적었다. 그리고 규칙스키마사이의 거리가 긴 수정된 시스템일수록 오류의 수가 늘어났을 뿐만 아니라 피실험자가 규칙스키마를 제대로 형성하지 못하여 혼동을 많이 일으키는 것으로 나타났다.

따라서 시스템의 일관성을 측정할 때에는 단지 규칙스키마의 개수뿐만이 아니라 규칙스키마 사이의 거리도 동시에 고려하여야 한다. 그리고 시스템을 설계할 때에는 단지 규칙스키마의 개수만을 줄이도록 노력하지 말고 이와 병행하여 규칙스키마 사이의 거리도 줄이도록 노력하여야 좀 더 일관성이 있는 시스템이 만들어 질 것이다.

참고문헌

- [1] A Dix, "HCI 2.0?: usability meets web 2.0", BCS-HCI '07 Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it, 2007, pp. 185-186
- [2] Anabela Samento, "Issues of Human Computer Interaction", 1st ed., IRM Press, 2005
- [3] C. Hsieh, M. Lu, E. Lin, "Consistency in Human-Computer Interfaces for End-Users", Organizational and End User Computing, 1994, pp. 3-10
- [4] Christopher D. Wickens, Justin G. Hollands, "Engineering Psychology and Human Performance", 3rd ed, Prentice-Hall, 2000
- [5] D. D. Woods, E. M. Roth, "Cognitive Engineering: Human Problem Solving with

Tools", Human Factors: The Journal of the Human Factors and Ergonomics Society, 1988, pp.415-430

- [6] D. J. Duke, P. J. Barnard, J. May, D. A. Duce, "Systematic development of the human interface", Proceedings in Software Engineering Conference, 1995, pp.313-321
- [7] J. Nielsen, "Coordinating user interfaces for consistency", ACM SIGCHI Bulletin, 1989, pp.63-65
- [8] John Dowell, John Long, "Target Paper: Conception of the cognitive engineering design problem", Ergonomics, 1998, pp.126-139
- [9] Judith Reitman Olson, Gary M. Olson, "The growth of cognitive modeling in human-computer interaction since GOMS", Human-Computer Interaction, 1990, pp.221-265
- [10] Muneo Kitajima, "A formal representation system for the human-computer interaction process", International Journal of Man-Machine Studies, 1989, pp.669-696
- [11] Stephen J. Payne, T.R.G. Green., "Task-action grammars: a model of the mental representation of task languages", Human-Computer Interaction, 1986, pp.93-133

저 자 소개



김 홍 규

1991: KAIST 경영과학과 학사
 1993: KAIST 산업공학과 석사
 2002: Purdue Univ. 산업공학과 박사
 1993 - 1999: LG생산기술원
 2002 - 2003: 삼성금융연구소
 2003 - 현재: 단국대학교 경영학부
 Email: heungkyu@dankook.ac.kr