Journal of the Ergonomics Society of Korea
Vol. 25, No. 2 pp.23-32, May 2006

윈도우 버튼 배열 구조의 사용성 평가

김태훈[1] · 한성호[2] · 권민정[3] · 이승희[4]

[1]LG전자 / [2]포항공과대학교 산업경영공학과 / [3]삼성전자 / [4]티맥스소프트

Usability Evaluation of a Window Button Arrangement

Taehun Kim[1], Sung H. Han[2], Minjung Kwon[3], Seunghee Lee[4]

[1]Mobile handset R&D center, LG Electronics, Seoul, 153-801
[2]Department of Industrial and Management Engineering, POSTECH, Pohang, 790-784
[3]Telecommunication Network Business, Samsung electronics, Seoul, 100-759
[4]Business Modeling Team, TmaxSoft R&D Center, Seongnam, 463-824

ABSTRACT

Window buttons located on a task bar provide information about windows that have been open as well as active. When the number of buttons on the task bar increases, the button size gets smaller and as a result, it is more difficult to identify. This study proposes new alternatives of presenting window buttons to search for a target button efficiently. An experiment was conducted to evaluate the usability of the new. Factors manipulated in the experiment include 'button arrangement methods' and 'order of buttons'. The 'button arrangement methods' factor is composed of three levels, 'one line', 'changeable line', and 'grouping buttons'. The 'order of buttons' factor is composed of three levels, 'fixed order', 'click-right', and 'order control by user'. The results showed that the main effects were significant in search time. In subjective satisfaction, the interaction effect was also significant. The results could be applied to the design of small screens such as PDA or cellular phones.

Keyword: Window buttons, Button arrangement, Usability evaluation

1. 서 론

GUI 환경에서 많은 수의 윈도우를 활성화 시켜놓고 작업을 수행하는 경우, 윈도우 간 이동을 편리하게 하거나, 원하는 윈도우를 검색하는 시간은 작업의 효율에 중요한 영향을 미친다. 실례로, 윈도우 간 이동하는 시간이 윈도우와 관련된 작업을 수행하는 시간의 63%를 차지하는 것으로 나타났으며(Kandogan and Sheiderman, 1997), 윈도우 수가 증가하면 검색 시간이 증가하기 때문에(Miah and Alty, 2000) 윈도우 관리 방법의 개선에 대한 연구가 필요하다.

사용하지 않을 경우, 윈도우의 크기가 작아지는 Vanishing window(Miah, 2000)를 이용하여 윈도우의 배치 및 크기를 조정하거나, 사용자에 따라 윈도우를 구조적으로 분류하는 Elastic window(Kandogan and Shneiderman, 1997) 등을 들 수 있다. 또한, 3D 공간에서의 윈도우 배치, 선택 등의 관리 방법(Dantzich et al., 2000)을 제시하는 연구가 있었으나, 윈도우 버튼에 대한 연구는 미흡하다.

윈도우 버튼은 사용 중인 응용 프로그램의 선택 상태와 작업 내용을 표시하는 버튼으로 작업 표시줄에 위치해 있으며, 사용자는 윈도우 버튼을 선택함으로써 원하는 윈도우로 전환할 수 있다. 현재 윈도우 버튼은 1열로 배열되는 것이 일반적이며, 윈도우 XP에서는 같은 응용 프로그램끼리 그룹으로 묶는 형태를 제공하고 있다. 1열로 배열되는 경우,

교신저자: 한성호
주 소: 790-784 경북 포항시 남구 효자동 산31, 전화: 054-279-2203, E-mail: shan@postech.edu

윈도우 버튼이 1열로 창을 띄운 순서대로 왼쪽에서 오른쪽으로 배열되며 윈도우의 개수가 증가함에 따라 윈도우 버튼의 크기는 줄어든다([그림 1] 참조). 이 경우, 윈도우 버튼의 순서가 고정되어 있어서 버튼의 위치를 기억해야 하며, 버튼의 수가 증가함에 따라 버튼의 크기가 줄어들기 때문에 검색 시간이 오래 소요되며(Fitt, 1953), 해당 버튼을 찾기가 어려워진다. 윈도우 버튼을 그룹으로 배열하는 경우, 그룹 내에서는 윈도우를 열어둔 순서대로 아래에서 위로 배열되며, 그룹 간의 버튼 순서는 변하지 않는다([그림 2] 참조). 버튼이 그룹으로 구성되어 있어서 그룹 버튼을 클릭해야 사용자가 원하는 버튼을 검색할 수 있기 때문에 해당 윈도우 버튼에 접근하기가 어려워진다.

기존 연구를 살펴보면 사무실의 시스템을 사용할 때 관심 주제나 분야가 바뀔 때마다 시스템의 정보를 재배열 하는 것이 사용자에게 편리하다는 연구 결과가 있으며(Malone, 1983), 메뉴의 검색에 관해서는, 화면의 구조화가 적절한 의미에 따라 이루어져 있을 경우, 정보를 검색하는 시간이 줄어들고(Hornof, 2001), 메뉴 항목을 카테고리에 의해 그룹화할 경우, 무작위로 나열한 메뉴들보다 검색 시간이 줄어든다는(Card, 1982) 결과가 있었다. 또한, 자주 선택하는 메뉴들이 분리하여 배치되어 있거나(Sears and Shneiderman, 1994), 사용자가 찾고자 하는 대상의 위치를 예측할 수 있을 경우(Hornof and Kieras, 1999), 검색 시간이 빨라진다. 윈도우 레이아웃 측면에서는 레이아웃의 일관성이 있을 경우, 시각적 인식, 위치의 기억 및 예측이 쉬워진다(Dantzich et al., 2000). 이러한 기존의 연구 결과들을 바탕으로, 기존 방식보다 사용성(Usability)을 향상시킬 수 있는 윈도우 버튼의 배열 구조를 제시하기 위해 다음과 같은 가설을 수립하였다.

1. 탐색에 필요한 정보의 양(버튼의 크기, 버튼 이름의 길이)을 늘리면 탐색의 수행도가 높아진다.

윈도우 버튼 배열의 수를 1열로 고정할 경우, 열어둔 윈도우의 개수가 증가할수록 윈도우 버튼의 크기가 작아져 윈도우 버튼에 나타나는 윈도우 이름에 대한 정보가 줄어 들게 된다. 만약 열어둔 윈도우의 수가 증가하여도 윈도우 버튼의 크기를 일정하게 고정하고 작업 표시줄의 윈도우 버튼 배열의 수를 늘린다면, 탐색에 필요한 정보의 양은 줄어들지 않는다. 또한 특정 버튼을 그룹으로 묶어 윈도우 버튼의 크기를 일정하게 유지한다면, 사용자가 볼 수 있는 해당 윈도우의 정보의 양은 줄어들지 않을 것이다. 따라서, 윈도우 버튼의 정보량을 증가시키면 버튼 검색 시간이 빨라질 것이다.

그림 1. 1열로 구성된 배열

2. 사용자가 최근에 사용하는 윈도우 버튼을 구분할 수 있으면 탐색의 수행도와 만족도가 높아진다.

컴퓨터 작업을 수행하다 보면, 열어둔 윈도우 중 몇 개의 윈도우는 계속 사용되는 반면, 몇 개의 윈도우는 잠시 사용되는 경우가 있다. 이들이 무작위로 배열되어 있는 경우가 많기 때문에 많은 버튼들 중에서 원하는 버튼을 찾는데 어려움이 따른다. 따라서, 자주 사용되는 윈도우를 반복적으로 이동하는 경우, 사용자가 버튼을 클릭할 때 마다 윈도우 버튼의 위치를 오른쪽으로 이동 시키면, 최근 사용되는 윈도우 버튼은 작업 표시줄의 오른쪽에 위치하게 되고, 자연스럽게 자주 사용하는 윈도우 버튼이 구분된다. 자주 사용하는 윈도우 버튼을 오른쪽에 배열하면, 버튼 검색 시간이 빨라지고, 만족도가 높아질 것이다.

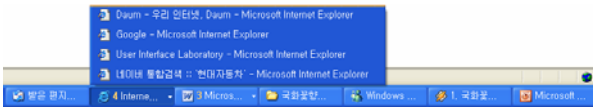

그림 2. 그룹으로 구성된 배열

3. 사용자가 윈도우 버튼의 위치를 직접 조작하면 탐색의 수행도가 높아진다.

사용자가 윈도우 버튼의 위치를 원하는 곳으로 옮길 수 있으면, 윈도우 버튼의 위치를 쉽게 기억할 수 있기 때문에 윈도우 버튼의 검색 시간을 줄일 수 있다.

본 연구에서는 윈도우 버튼의 배열 구조를 버튼 배열 방식과 배열 순서로 구분하여 정의하고, 위의 가설에 따라 다양한 윈도우 버튼의 배열 구조를 제시하고 사용성 실험을 통하여 평가하였다.

2. 실험 방법

2.1 준비 조사

윈도우 버튼에 대한 사용자의 사용 행태와 윈도우 버튼의 개선 방안에 대한 선호도를 조사하기 위하여 14명의 대학원생을 대상으로 설문 조사를 수행하였다. 설문 내용은 윈도우 사용 중의 윈도우 버튼의 검색 및 인식의 문제점과 사용하는 윈도우 버튼 방식에 대한 의견 및 문제점, 제안된 윈도우 버튼 방식에 대한 선호도 등이다. 열어둔 윈도우 개수와 평소 사용하는 응용 프로그램의 종류에 대하여 알아보기 위해 위의 선호도 설문 조사와 별개로 조사를 수행하였으며, 본 연구가 이루어진 대학 내에서 컴퓨터를 사용하고 있는 사람을 무작위로 7명을 선정하여 사용자가 사용하는 윈도우 수와

응용 프로그램을 조사하였다. 설문 결과, 사용 빈도는 윈도우 버튼의 순서를 직접 이동 시킬 수 있는 방식과 윈도우 버튼이 최근에 활성화된 순서대로 배열되는 방식, 그리고 윈도우 버튼을 그룹으로 조작하여 사용하는 방식은 50% 이상의 선호도를 나타내었다. 또한, 사용하는 윈도우 버튼의 크기를 크게 하고, 사용하지 않으면 윈도우 버튼의 크기를 작게 하는 방식에 대해서는 42.86%만이 긍정적인 응답을 보였으나 이는 다른 방식보다 부정적인 측면이 많으므로, 연구 범위에서 제외되었다. 최근 사용한 윈도우 버튼의 위치에 대한 설문에서는 작업 표시줄의 오른쪽에 위치하여야 한다는 답변이 57.14%이며, 그룹 내에서는 아래쪽에 위치하는 것을 선호하는 답변이 100%였다. 평소 열어둔 윈도우의 개수는 평균 9.14개 이며, 표준 편차는 2.36개로 나타났다. 응용 프로그램의 종류는 [표 1]과 같이 다양하게 나타났으며, 사용자는 인터넷을 많이 열어두는 것을 알 수 있다. 사용 빈도는 사용자가 열어두는 응용 프로그램 개수의 평균이다.

표 1. 응용 프로그램 사용 빈도 결과(개수)

프로그램 종류	PDF	음악	메신저	워드	파워 포인트	엑셀	인터넷	메일	폴더
사용 빈도	0.14	0.14	1.29	1.14	1.00	0.14	3.57	0.71	1.00

2.2 피실험자

본 실험에는 20세에서 28세 사이(평균: 23.2, 표준 편차: 2.2)의 대학생 및 대학원생 18명(남: 9명, 여: 9명)이 피실험자로 참가하였다. 윈도우를 사용하는데 어려움이 없고, 윈도우 사용에 익숙한 사용자를 선정하기 위하여 컴퓨터 사용 경력이 최소 3년 이상인 사람을 대상으로 실험을 수행하였다. 또한 마우스를 사용하는데 어려움이 없는 오른손잡이로 제한하였다.

2.3 실험 장비

실험 장비는 Window XP 운영체제를 탑재한 컴퓨터와 해상도가 1024 X 768인 20인치 CRT모니터를 사용하여 실험을 수행하였다. Visual C++을 이용하여 실험 환경을 구현하였으며, 독립 변수 외의 기타 외부 요인들의 영향을 줄이기 위해 윈도우에서 사용되고 있는 아이콘을 그대로 사용하였다. 해당 윈도우 버튼에 커서를 위치시키면, 윈도우 버튼의 전체 이름을 제공하는 Tool-tip 기능도 제공하였다. 또한, 윈도우 버튼의 위치와 크기 및 레이블도 MicroSoft사의 윈도우와 동일하게 구성하였다.

평소 열어두는 윈도우 버튼의 개수(평균 9.14개, 표준 편차 2.36개)의 조사 결과를 바탕으로 실험에서는 열어둔 윈도우 버튼의 최대 개수인 12개로 구성하였다. 열어놓은 12개 윈도우의 응용 프로그램의 종류는 준비 조사 결과([표 1] 참조)를 바탕으로 선정하였으며, 각 응용 프로그램의 개수는 사용 빈도와 작업의 흐름을 고려하여 설정하였다. 응용 프로그램의 개수는 폴더 2개, MS-word 3개, PowerPoint 1개, Acrobat reader 1개, 인터넷 웹사이트 3개, 메신저 1개, Winamp 1개로 구성되었다.

제공되는 윈도우 버튼의 개수와 종류에 따른 영향을 최소화하기 위하여 피실험자에게 제공되는 윈도우 버튼의 개수와 응용 프로그램의 종류는 항상 같게 하였고 윈도우 버튼의 배열 순서를 무작위로 구현하였다. 그룹 배열 시, 순서 조작의 경우를 제외하고, 그룹 내에서 제공되는 응용 프로그램의 종류와 개수가 같다. 그룹 방식은 같은 응용 프로그램끼리 그룹화하여 나타내었으며, 총 7개의 그룹으로 구성되었다. 그룹 방식의 순서 조작의 경우, 그룹의 개수와 그룹 내의 응용 프로그램은 사용자가 임의대로 조작할 수 있는 방법으로 설계하였다.

2.4 실험 계획

본 논문에서는 여러 응용 프로그램의 윈도우 버튼을 이동하는 실험을 수행하였다. 이동 경로는 여러 응용 프로그램으로 이동하는 경로를 나타내며, 윈도우 버튼의 이동은 12개의 응용 프로그램 중 MS-word와 PowerPoint에 제시된 이동 경로에 따라 수행된다. 해당 응용 프로그램의 이동은 마우스를 이동하여 응용 프로그램의 윈도우 버튼을 클릭함으로써 수행된다. 여러 응용 프로그램으로 이동하는 작업을 반복적으로 수행하여 12개의 윈도우 버튼을 이동한다. 윈도우 버튼을 검색하는 시간은 윈도우 버튼의 시각적 검색 시간과 마우스 포인팅 시간의 합으로 정의하였으며(Anderson et al, 1997), 피실험자가 윈도우 버튼을 검색하는 시간만을 측정하기 위해서 프로그램의 오른쪽 하단에 "Start" 버튼과 "Stop" 버튼을 배치하였다.

피실험자는 무작위로 배열된 12개 윈도우 버튼을 정해진 순서에 따라 순차적으로 클릭하는 윈도우 버튼 이동 작업을 한 실험 조건당 50~70회 수행하였다. 윈도우 버튼 이동 작업 1회는 '12개 윈도우 버튼의 종류 및 위치 확인 → 윈도우 버튼 이동 경로 숙지 → 'Start' 버튼을 이용한 작업 시작 → 제시된 이동 경로에 따라 12개 윈도우 버튼들 간의 이동 작업 수행 → 'Stop' 버튼을 이용한 작업 종료' 등 4단계의 순서로 진행되었다.

정해진 이동 경로로 이동되지 못한 경우, 에러로 기록하였다. 작업에서 학습의 영향을 줄이기 위해, 실험 조건마다 윈도우의 이동 경로를 다르게 구성하였으며, 버튼 간 무작위로

이동하기 때문에 실험 조건마다 이동 횟수가 다르게 구성되었다. 수행 횟수에 따른 결과를 일반화하기 위하여 피실험자는 실험 조건 당 버튼을 50~70회 이동하였으며, 검색 시간은 횟수 당 평균 시간으로 측정하였다. 실제 윈도우를 사용할 때, 2개의 윈도우 버튼을 반복적으로 이동하는 경우가 많이 나타나므로, 이를 고려하기 위하여 지시사항에 두 개의 특정 버튼을 각각 5번씩 반복적으로 윈도우를 이동하는 경우를 포함시켰다.

본 실험에서는 윈도우 버튼의 배열 방식과 배열 순서를 독립 변수로 사용하였다. 버튼의 배열 방식과 배열 순서에 대한 수준과 정의는 [표 2]와 같다.

표 2. 독립 변수의 수준 정의

독립변수	수준	정 의
윈도우 버튼의 배열 방식	1열 배열 방식	윈도우 버튼이 1열에 배열되는 방식으로 윈도우 개수가 증가하면, 윈도우 버튼의 크기가 줄어드는 기존 방식([그림 3] 참조)
	가변 배열 방식	윈도우 버튼의 크기를 고정시키는 방법으로, 윈도우 개수가 증가하면 작업표시줄의 열 개수가 자동으로 증가하는 방식([그림 4] 참조)
	그룹 배열 방식	윈도우 버튼들을 그룹화하여 배치하는 방식으로, 원하는 윈도우 버튼이 있는 그룹을 선택한 후 그룹 내의 윈도우 버튼을 클릭하는 방식([그림 5] 참조)
윈도우 버튼의 배열 순서	순서 고정 방식	윈도우를 실행한 순서대로 윈도우 버튼이 왼쪽에서 오른쪽으로 배열되는 기존의 방식
	최근 순서 방식	사용자가 윈도우를 활성화시키면 활성화된 윈도우 버튼이 작업 표시줄의 오른쪽 끝으로 이동되어, 윈도우 버튼이 최근 사용한 순서대로 오른쪽부터 배열되는 방식
	순서 조작 방식	사용자가 윈도우 버튼을 직접 이동하여 윈도우 버튼의 순서를 재배열하는 방식

그림 3. 1열 배열 방식의 예

그림 4. 가변 배열 방식의 예

그림 5. 그룹 배열 방식의 예

본 연구의 종속 변수로 윈도우 버튼을 선택할 때 소요되는 작업 수행 시간(Task completion time) 및 버튼의 반복 이동 시 버튼을 선택할 때 소요되는 작업 수행 시간, 에러 횟수(Number of error) 등의 객관적 수행도와 직관성(Intuitiveness), 편리성(Convenience), 전반적 만족도(Overall Satisfaction) 등의 주관적 만족도를 고려하여 윈도우 버튼의 사용성을 평가하였다([표 3] 참조). 작업 수행 시간은 평균 검색 시간으로 측정함으로써, 각 조건 별로 수행된 횟수와 상관없이 윈도우 버튼을 클릭하는데 소요되는 시간을 비교할 수 있다. 에러 횟수는 제시된 윈도우 버튼을 클릭하지 못하였을 경우로 정의하였다. 실험 계획은 Within subjects design을 이용하였으며, Balanced Latin Square Design을 이용해 실험 순서를 결정하였다.

표 3. 종속 변수의 정의

	종속 변수	정의 및 평가 방법
객관적 수행도	작업 수행 시간	윈도우 버튼을 클릭하는데 소요되는 시간을 윈도우 버튼을 클릭한 횟수로 나눈 평균 검색 시간
	반복 이동 시 작업 수행 시간	두 개의 특정 윈도우 버튼을 반복적으로 이동하는 경우의 평균 검색 시간
	에러 횟수	윈도우 버튼을 잘못 선택한 횟수
주관적 만족도	직관성	윈도우 버튼의 위치를 직관적으로 예측 가능한 정도로 정의하며, 0에서 100점 사이의 점수로 측정
	편리성	편리성은 윈도우 버튼의 선택과 조작 과정을 포함하여 버튼의 배열 구조의 편리한 정도로 정의하며, 0에서 100점 사이의 점수로 측정
	전반적 만족도	만족도는 윈도우 버튼을 사용하는 동안 전반적으로 느끼는 만족 정도로 정의하며, 0에서 100점 사이의 점수로 측정

2.5 실험 절차

실험 절차는 실험 소개, 훈련, 본 실험, 질의의 4단계로 진행되었다. 실험 소개 단계에서는 피실험자가 신상 정보와 윈도우 사용 형태에 대한 간단한 설문에 응답하고, 피실험자에게 실험 목적 및 방법을 숙지시켰다. 훈련 단계에서는 피실험자가 9가지 실험 조건에 대해 익숙해지고, 윈도우 버튼의 사용 방법을 숙지할 수 있도록 각 실험 조건을 5분씩 총 45분 동안 수행하였다. 훈련을 마치면 10분의 휴식 시간을 제공하였다. 본 실험 단계에서는 각 피실험자가 서로 다른 9가지 실험 조건에서 주어진 작업을 수행하였으며, 각 실험 조건의 실험이 끝날 때마다 직관성, 편리성, 전반적 만족도에 대해 평가하였다. 본 실험이 끝난 후, 피실험자에게 가장

선호하는 방식과 기타 방식에 대한 느낌이나 기타 의견에 대하여 설문 조사를 수행하였다. 총 실험 시간은 2시간 30분이 소요되었으며, 각 실험 사이에 중간 휴식 시간을 2~3분 정도 제공하였다. 실험의 진행이 어려운 경우, 실험 진행을 위하여 피실험자에게 충분한 휴식을 제공하였다.

3. 실험 결과

평균 검색 시간과 주관적 만족도 요소들에 대해 분산 분석을 실시하였으며, 그 결과를 요약하면 [표 4]와 같다. 편리성에서 배열 방식에 의한 차이와 객관적 수행도에서 실험 인자 간의 교호 작용은 유의하지 않은 결과가 나타났으며, 그 외에는 유의 수준 5%에서 모두 유의한 결과가 나타났다.

표 4. 항목별 ANOVA 결과(p-value)

	작업 수행 시간	반복 이동 시 작업 수행 시간	직관성	편리성	전반적 만족도
배열 방식 (A)	<0.01*	<0.01*	0.01*	0.06	<0.01*
배열 순서 (B)	<0.01*	0.04*	<0.01*	<0.01*	<0.01*
A × B	0.21	0.98	<0.01*	<0.01*	<0.01*

3.1 객관적 수행도

3.1.1 작업 수행 시간

버튼 배열 방식과 배열 순서의 실험 인자 각각에 의한 차이는 유의했던 반면, 실험 인자 간의 교호 작용은 유의한 차이가 없었다. [그림 6]은 버튼 배열 방식 간 작업 수행 시간의 주 효과 분석 결과이다. SNK test 결과, 버튼 배열 방식에서 그룹 배열(1.759초)이 가장 긴 시간이 소요된 것으로 나타났으며, 가변 배열(1.215초)과 1열 배열(1.228초)간에는 유의한 차이가 없었다.

[그림 7]은 배열 순서 간 작업 수행 시간의 SNK test 결과이다. 최근 순서 방식(1.518초)의 검색 시간이 가장 많이 소요되었으며, 순서 고정 방식(1.396초)과 순서 조작 방식(1.289초)순으로 시간이 적게 소요되었다. 이들 세 수준 간에 유의한 차이가 존재하였다.

3.1.2 반복 이동 시 작업 수행 시간

반복 이동 시 작업 수행 시간에 대한 버튼 배열 방식, 배열 순서에 의한 차이는 각각 유의했던 반면, 실험 인자 간의 교호 작용은 유의한 차이가 없었다. [그림 8]은 두 버튼의

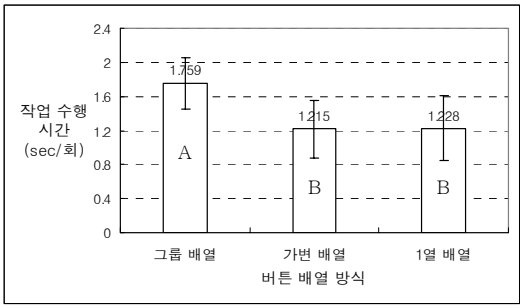

그림 6. 버튼 배열 방식 간 작업 수행 시간 비교(그림에서 서로 다른 문자간에는 통계적으로 (α=0.05) 유의함)

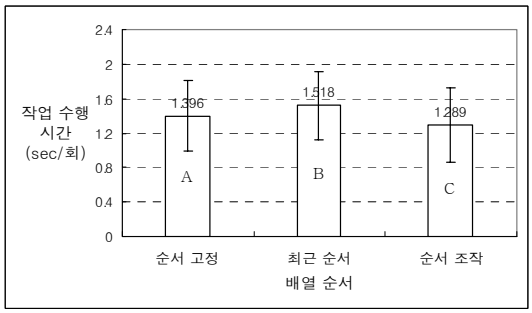

그림 7. 배열 순서 간 작업 수행 시간 비교(그림에서 서로 다른 문자간에는 통계적으로(α=0.05) 유의함)

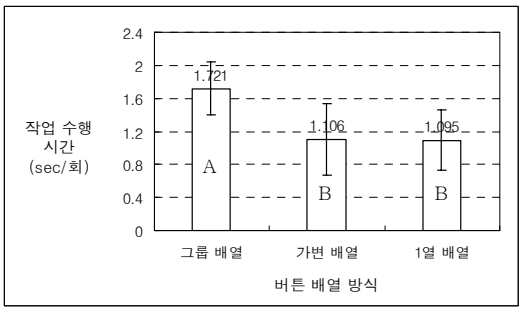

그림 8. 두 버튼의 반복 이동 시 버튼 배열 방식 간 작업 수행 시간 비교(그림에서 서로 다른 문자간에는 통계적으로(α=0.05) 유의함)

반복 이동 시 버튼 배열 방식 간 작업 수행 시간의 SNK test 결과이다. 버튼 배열 방식에서는 그룹 배열 방식(1.721초)이 1열 배열(1.106초)이나 가변 배열 방식(1.095초)보다 시간이 오래 소요되었으며, 그룹 배열 방식과 가변 배열 간 그리고 그룹 배열과 1열 배열 간에 유의한 차이가 발생하였다.

[그림 9]는 두 버튼의 반복 이동 시 배열 순서의 SNK test 결과이다. 배열 순서 간 작업 수행 시간을 비교해 보면, 순서 조작 방식(1.204초)이 순서 고정 방식(1.383초)이나

최근 순서 방식(1.335초)에 비해 시간이 적게 소요되는 것으로 나타났으며, 최근 순서 방식과 순서 조작 방식 간에 유의한 차이가 있었다.

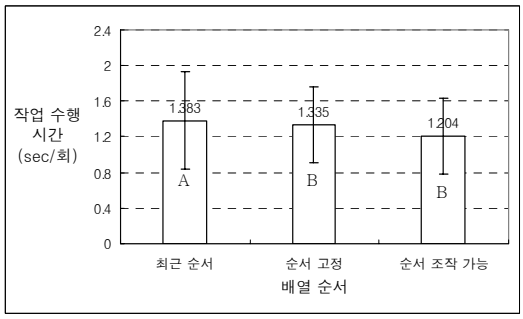

그림 9. 두 버튼의 반복 이동 시 배열 순서 간 작업 수행 시간(그림에서 서로 다른 문자간에는 통계적으로(α=0.05) 유의함)

각 실험 조건에서 평균 에러 회수는 2회 이하로 매우 적었다(평균 1.52회). 각 실험 조건에서의 에러 회수를 Chi-square test를 이용해 비교한 결과, 에러 회수의 차이는 유의하지 않은 것으로 나타났다($6.15 < (\chi^2(0.95, 4) = 9.49)$).

3.2 주관적 만족도

3.2.1 직관성

직관성에 대한 버튼 배열 방식과 배열 순서, 버튼 배열 방식과 배열 순서의 교호 작용이 모두 유의한 차이를 나타내었다([표 4] 참조). SNK test 결과, 1열 배열 방식(57.5)이 그룹 배열 방식(65.5)과 가변 배열 방식(62.9)에 비해 직관성이 떨어지는 것으로 나타났다. 배열 순서에서는 순서 조작 방식(74.4)과 최근 순서 방식(60.6)이 순서 고정 방식(50.9)에 비해 높은 직관성을 갖는 것으로 나타났다.

직관성의 교호 작용에 대한 Simple effect test를 수행한 결과, 순서 고정 방식에서 세 가지 버튼 배열 방식에 따라 직관성에 유의한 차이가 있는 것으로 나타났으며, 최근 순서 방식 및 순서 조작 방식에서는 버튼 배열 방식 간에 유의한 차이가 없었다([그림 10] 참조). 순서 고정 시 버튼 배열 방식 간 SNK test를 수행한 결과, 그룹 배열(63.2)이 가장 높게 평가되었으며, 가변 배열(49.5), 1열 배열(40.1) 간 세 수준에서 유의한 차이가 나타났다.

3.2.2 편리성

편리성의 경우, 배열 순서와 버튼 배열 방식과 배열 순서의 교호 작용이 모두 유의한 차이를 나타내었다([표 1] 참조). SNK test 결과, 배열 순서에서는 순서 조작 방식(69.0)이 최근 순서 방식(58.8)과 순서 고정 방식(48.0)에 비해

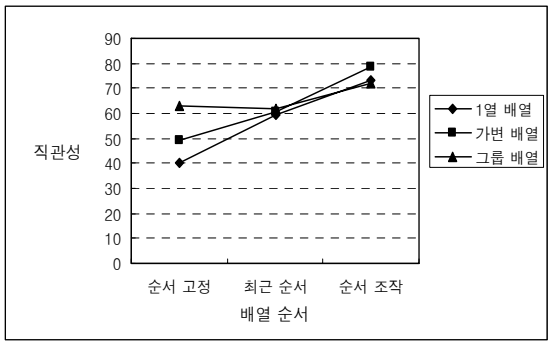

그림 10. 직관성의 교호 작용 분석

편리성이 높은 것으로 나타났으며, 세 수준에서 유의한 차이가 발생하였다. 편리성의 교호 작용에 대한 Simple effect test를 수행한 결과, 순서 고정 방식과 순서 조작 방식에서 버튼 배열 방식이 유의한 차이가 나타났다([그림 11] 참조).

순서 고정 시와 순서 조작 시의 버튼 배열 방식 간 SNK test를 수행한 결과, 순서 고정 시에는 그룹 배열(60.4)이 가장 높은 결과를 나타내었으며, 가변 배열(46.9), 1열 배열(36.8)과 유의한 차이를 나타내었다. 순서 조작 시의 버튼 배열 방식 간의 SNK test 결과, 가변 배열(78.3)이 가장 높은 값을 나타내었으며, 다른 배열 방식과 유의한 차이를 나타내었다.

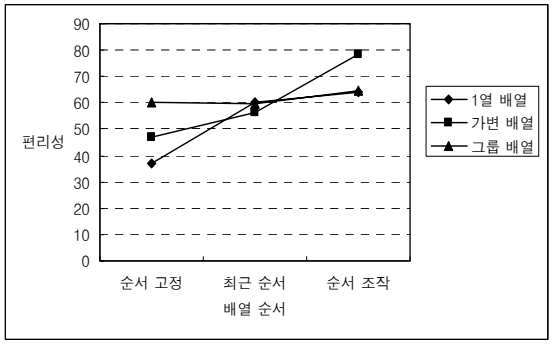

그림 11. 편리성의 교호 작용 분석

3.2.3 전반적 만족도

전반적 만족도에 대한 버튼 배열 방식과 배열 순서, 버튼 배열 방식과 배열 순서의 교호 작용이 모두 유의한 차이를 나타내었다([표 1] 참조). SNK test 결과, 그룹 배열 방식(63.1)과 가변 배열 방식(61.8)이 1열 배열 방식(53.0)에 비해 전반적 만족도가 높은 것으로 나타났다. 배열 순서에서는 순서 조작 방식(70.7)이 전반적 만족도가 가장 높게 나타났으며, 최근 순서 방식(59.0), 순서 고정 방식(49.1)

순으로 높게 나타났다. 배열 순서에서는 세 수준 간에 유의한 차이가 있었다.

전반적 만족도의 교호 작용에 대한 Simple effect test를 수행한 결과, 순서 고정 방식과 순서 조작 방식에서 버튼 배열 방식 간 유의한 차이가 나타났다([그림 12] 참조). 순서 고정 시와 순서 조작 시의 SNK test를 수행한 결과, 순서 고정 시 그룹 배열 방식(61.4)이 가장 높은 결과를 나타내었으며, 그룹 배열, 가변 배열(48.6), 1열 배열(37.4)의 세 수준 간에 유의한 차이가 발생하였다. 순서 조작 시 가변 배열(78.3)과 그룹 배열(68.3) 간 그리고 가변 배열과 1열 배열(65.4) 간 유의한 차이를 나타내었으며, 가변 배열(78.3)이 전반적 만족도 측면에서 가장 높은 결과를 나타내었다.

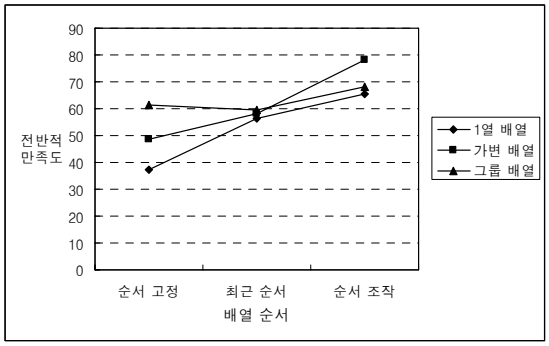

그림 12. 전반적 만족도의 교호 작용 분석

4. 토 의

4.1 버튼 배열 방식

4.1.1 그룹 배열 방식

그룹 배열 방식은 버튼이 포함되어 있는 그룹을 찾고 그룹 내에서 해당 버튼을 찾아야 하기 때문에 가변 배열과 1열 배열보다 시간이 많이 소요된 것으로 판단된다([표 5] 참조). 반면에, 윈도우 버튼을 그룹을 통해 구조적으로 분류할 수 있어서 직관성과 전반적 만족도측면에서 그룹 배열 방식에 대한 사용자의 선호도가 높게 나타났다.

피실험자의 사용 행태를 관찰한 결과, 순서 조작 시 피실험자는 12개의 버튼을 chunking을 통해 3~4개의 그룹으로 구성하여, 그룹 간 버튼 검색을 편리하게 하려는 경향이 나타났다(Wickens and Hollands, 1999). 사용자는 그룹 간 검색 시간을 단축하기 위하여, 그룹을 단순히 하려는 경향을 보였다. 또한, 그룹 내에서 사용자가 원하는 대로 정보를 그룹화하였기 때문에 사용자의 user model과 일치하여 검색의 효율성을 향상시킨 것으로 판단된다. 그러나, 버튼을 두 번 누르게 되는 문제가 발생하며, 접근성 측면의 개선이 필요하다.

4.1.2 1열 배열 방식

1열 배열 방식은 열어둔 버튼의 수가 증가함에 따라, 버튼의 크기가 작아지고, 버튼에서 제시되는 정보의 양은 매우 제한된다. 따라서 사용자들은 제한된 정보로 인해 버튼을 검색하는 시간이 오래 소요되었다. 윈도우의 제목이 비슷한 두 개의 버튼이 존재할 경우, 버튼의 크기로 인하여 제목의 앞 부분만이 제공되어 사용자는 제목이 비슷한 여러 윈도우 버튼을 이동하거나, Tool tip을 확인하는 등의 불필요한 행동을 유발하기 때문에 수행 시간이 오래 소요된다. 버튼에 나타나는 정보가 제한되지 않도록 윈도우 버튼을 3~4개 정도 열어둔다면 수행도 면에서 1열 배열 방식이 다른 배열 방식과 차이가 없을 것으로 판단된다. 주관적 만족도 측면에서 새로운 방식에 대한 호기심과 편리성으로 인해 1열 배열 방식이 낮은 평가를 얻었다.

4.1.3 가변 배열 방식

가변 배열 방식은 윈도우 버튼이 일정한 크기를 유지하기 때문에 1열 배열 방식에 비해 더 많은 정보를 제공할 수 있다. 따라서 버튼 정보의 인식이 용이하며, 버튼의 검색 시간이 적게 소요되었다. 가변 배열 방식은 1열 배열 방식보다 버튼을 검색하는 공간이 늘어나기 때문에 검색 시간이 증가된다. 그러나, 버튼에서 제시되는 정보의 양에 의해 버튼을 선택하기 용이하므로, 검색 시간이 단축된 것으로 판단된다.

본 실험은 윈도우 버튼의 검색을 고려하였기 때문에 윈도우 버튼의 크기에 따른 윈도우 작업의 영향을 판단하기는 어려웠다. 질의 결과, 사용자는 가변 배열에 만족하는 것으로 나타났으나, 윈도우에서 행하여 지는 여러 작업(인터넷 검색, 문서 작성, 읽기 작업 등)을 고려한다면, 주 작업 공간의 제약으로 인해 가변 배열의 선호도는 많이 줄어들 것으로 판단된다.

4.2 배열 순서

4.2.1 순서 고정 방식

순서 고정 방식은 여러 윈도우로 작업을 수행할 때, 사용자에게 버튼의 위치를 기억해야 하는 부담을 주며, 검색 시간이 오래 소요된다. 그러나, 검색 버튼의 개수가 적을 경우, 사용자는 버튼의 위치를 기억하기 쉬울 것으로 판단된다. 피실험자의 행동을 관찰한 결과, 피실험자는 윈도우 버튼을 오른쪽에서 왼쪽 방향으로 검색하는 경향이 있다. 이는 왼쪽에서 오른쪽 방향으로 진행되는 visual search 방식(Wic-

kens and Hollands, 1999)과 상반된 결과를 나타내었으며, 일반적으로 최근에 열어둔 윈도우 버튼을 찾는 경향을 나타내었다.

4.2.2 순서 조작 방식

순서 조작 방식은 사용자가 원하는 대로 윈도우 버튼의 순서를 배열할 수 있기 때문에 사용자의 윈도우 버튼의 위치에 대한 기억 및 예측을 가능하게 한 것으로 판단된다. 따라서, 객관적 수행도와 주관적 만족도 측면에서 다른 방식보다 좋은 결과를 나타내었다. 이는 버튼의 위치를 옮기는 주체가 시스템보다 사용자 자신일 경우, 좋은 결과를 나타낸 것으로 판단되며, 사용자가 임의로 조작할 수 있는 방식이 시스템에 의해 메뉴를 적응시키는 방법보다 나은 결과를 보여준다(Findlater, 2004). 일반적으로 사용자가 시스템을 사용할 때, 관심 주제나 분야가 바뀔 때마다 정보를 재배열 하는 것이 편리하며(Malone, 1983), 여러 응용 프로그램을 지속적으로 사용해야 하는 작업자의 경우, 순서 조작 방식은 더 유용하게 사용될 수 있을 것으로 기대된다.

4.2.3 최근 순서 방식

최근 순서 방식은 윈도우 버튼이 자주 옮겨지는 방식이어서 사용자들의 혼란을 초래하는 방식으로 평가되었다. 그러나, 몇 개의 윈도우 사이를 반복적으로 전환하는 작업의 경우, 자주 사용하는 윈도우 버튼이 오른쪽에 모여 있어 위치를 빠르게 파악할 수 있다는 장점이 있다. 실험 후의 질의 시간에서 피실험자 18명 중 3명이 이 방식을 가장 선호하였으며, 이 방식이 익숙해지면 평균 검색 시간이 다른 방식보다 적게 소요될 수 있고, 사용하기 편리할 것으로 기대하였다.

4.3 교호 작용

교호 작용에서 배열 순서를 고정시켰을 경우, 버튼 배열에 따라 사용자의 주관적 만족도 면에서 차이가 나타나는 것을 알 수 있다([표 5]) 참조). 순서 고정 방식의 경우, 사용자는 그룹 배열 방식을 통해 그룹으로 윈도우 버튼을 구조적으로 분류하여 버튼의 위치 및 내용을 기억하는 부담을 줄여주는 방식을 선호하는 것으로 나타났다. 그러나, 순서 조작 방식의 경우, 사용자가 윈도우 버튼의 위치를 사용자 임의대로 조작하여 버튼의 위치를 쉽게 기억할 수 있기 때문에, 사용자는 윈도우의 내용을 확인하기 용이한 가변 배열 방식을 선호하는 것으로 나타났다.

사용자 설문 조사 결과, 가변 배열의 순서 조작 방식이 수행도와 만족도 측면에서 가장 좋은 결과를 보였으며, 피실험자가 가장 선호하는 방식으로 평가되었다. 사용자가 윈도우 버튼을 검색하는 과정을 확인한 결과, 대부분의 사용자들은 사용자의 기억에 의지하여 윈도우 버튼의 위치를 대략적으로 찾게 되며, 대략적인 위치에 포함된 윈도우 버튼의 이름을 확인하여 버튼을 검색하는 것으로 나타났다. 따라서 가변 배열의 순서 조작 방식은 버튼의 내용과 위치를 쉽게 기억하게 하며, 사용자에게 기억의 부담을 줄여주는 것으로 판단된다.

4.4 설계 지침 요약

본 연구를 통해 분석된 실험 결과를 요약하면 [표 5]와 같다. 표에서 부등호는 좋은 결과를 보여준 순서이며, 등호는 두 인자 간 유의한 차이가 없음을 의미 한다.

버튼 배열 방식으로는 가변 배열 형식이 그룹 배열이나 1열 배열 방식보다 작업 수행도와 만족도 측면에서 모두 우수한 것으로 분석되었다. 또한, 윈도우 버튼 배열 순서로는 사용자가 윈도우 버튼의 위치를 조작할 수 있는 방식이 최근 사용 버튼을 구분하여 제시하는 방식과 순서를 고정하는 방식보다 높은 수행도와 높은 만족도를 나타내었다 ([표 5] 참조). 사용자가 버튼 위치와 순서를 직접 설정할 수 있는 방식은 사용자가 버튼의 위치를 기억하기 용이하기 때문으로 판단된다.

따라서, 버튼 탐색 수행도와 주관적 만족도를 향상시키기 위해서는 윈도우 버튼의 위치를 사용자가 임의로 변경할 수 있어야 하며, 이때 윈도우 버튼들을 가변 배열 방식으로 제

표 5. 실험 결과 요약

버튼 배열 방식	작업 수행 시간	그룹 배열 < 가변 배열 = 1열 배열
	직관성	1열 배열 < 가변 배열 = 그룹 배열
	전반적 만족도	1열 배열 < 가변 배열 = 그룹 배열
배열 순서	작업 수행 시간	최근 순서 < 순서 고정 < 순서 조작
	두 버튼의 반복 이동 시 작업 수행 시간	최근 순서 < 순서 조작
	주관적 만족도	순서 고정 < 최근 순서 < 순서 조작
버튼 배열 방식 × 배열 순서	순서 고정 시, 버튼 제시 방식 간 직관성	1열 배열 < 가변 배열 < 그룹 배열
	순서 고정 시, 버튼 제시 방식 간 편리성	1열 배열 = 가변 배열 < 그룹 배열
	순서 고정 시, 버튼 제시 방식 간 전반적 만족도	1열 배열 < 가변 배열 < 그룹 배열
	순서 조작 시, 버튼 제시 방식 간 편리성	1열 배열 = 그룹 배열 < 가변 배열
	순서 조작 시, 버튼 제시 방식 간 전반적 만족도	1열 배열 = 그룹 배열 < 가변 배열

시할 수 있도록 시스템을 구현하는 것이 요구된다. 즉, 사용자에게 윈도우 버튼의 위치, 순서, 배열 형태 등을 임의로 설정할 수 있도록 하는 것이 사용성 측면에서 추천된다.

그러나, 윈도우 버튼의 위치를 사용자가 임의로 변경할 수 있도록 하는 것이 시스템 구현 상 불가능할 경우에는, 그룹 배열 형태로 유사한 형태와 특징을 갖는 윈도우 버튼들을 그룹으로 제시하는 것이 만족도 측면에서 유리하다.

5. 결 론

본 연구에서는 일반적인 PC 작업 환경을 대상으로 하여 새로운 윈도우 버튼의 배열 구조를 제시하였다. 사용성 평가 결과, 버튼의 순서를 사용자가 직접 조작할 수 있으며, 버튼의 이름을 쉽게 볼 수 있는 크기로 윈도우 버튼의 크기를 고정시킨 방식이 수행도와 만족도 측면에서 사용성을 향상시키는 것으로 나타났다. 따라서, 일반적인 PC 작업 환경에 윈도우 버튼의 순서 조작 기능과 이름을 쉽게 읽을 수 있도록 버튼의 크기를 고정하면 수행도와 만족도를 동시에 향상시킬 수 있을 것이다.

본 연구는 윈도우 관리 측면에서 윈도우의 버튼을 대상으로 한 실험이며, 윈도우 버튼의 배열 구조를 개선함으로써 윈도우 작업을 효율을 높일 수 있음을 보였다는 점에서 의의를 갖는다. 순서 조작 방식은 사용자가 임의대로 버튼을 옮길 수 있는 방식이므로 화면이 작은 경우에서 Personalization을 위한 방안으로 사용될 수 있으며, 메뉴 설계 시, 순서 조작 가능한 메뉴를 제시하여, 검색의 효율성을 증대시킬 수 있을 것이다.

본 논문은 사용자의 행동 패턴을 감안한 menu 방식으로 최근 순서 방식만을 제시하였으며, 사용자가 원하는 메뉴가 시스템에 의해 정확한 시간에 적절한 방법으로 제시되는 다른 사용자의 행동 패턴을 감안한 menu 방식을 고려할 필요가 있다. 따라서, 앞으로 사용자의 작업 패턴을 분석하는 알고리즘을 개발하여 사용자가 원하는 다음 작업을 예측하는 인터페이스를 개발한다면 좀 더 효율적인 윈도우 버튼 구조의 제시가 가능할 것이다.

참고 문헌

Card, S. K., "User perceptual mechanisms in the search of computer command menus", Proceedings of conference on Human factors in computer systems, pp. 190-196, Gaithersburg. Maryland. 1982.

Findlater, L. and McGrenere, J., "A comparison of static, adaptive, and adaptable menus.", Proceedings of ACM CHI 2004. pp. 89-96, Vienna, Austria. 2004.

Fitts, P. M., The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391. 1954.

Hornof, A. J. and Kieras, D. E., "Cognitive modeling demonstrates how people use anticipated location knowledge of menu items", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 410-417, Pittsburgh. PA. 1999.

Hornof, A. J., Visual Search and Mouse-Pointing in Labeled versus Unlabeled Two-Dimensional Visual Hierarchies, A*CM Transactions on Computer-Human Interaction*, 8(3), 171-197, 2001.

Kandogan, E. and Shneiderman, B., "Elastic Windows: evaluation of multi-window operations", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 250-257, Atlanta. Georgia. 1997.

Malone, T. W., How do people organize their desk? Implications for the design of office automation systems, *ACM Transaction on office information systems*, 1(1), 99-112, 1983.

Miah, T. and Alty, J., Vanishing Windows: a technique for adaptive window management, *Interacting with Computers*, 12(4), 337-355, 2000.

Robertson, G., Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D. and Gorokhovsky, V., "The Task Gallery: a 3D window manager", Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 494-501, The Hague. The Netherlands. 2000.

Sears, A. and Shneiderman, B., Split menus: effectively using selection frequency to organize menus, A*CM Transactions on Computer-Human Interaction*, 1(1), 27-51, 1994.

Wichens, C. D. anf Hollands, J. G., Engineering psychology and human performance, 3rd edition, Prentice Hall, New Jerssey, 1999.

◐ 저자 소개 ◑

❖ 김 태 훈 ❖ nandabari@lge.com
포항공과대학교 산업공학과 석사
현 재: LG전자 단말연구소 주임연구원
관심분야: 사용자 인터페이스 개발 및 평가, 사용성 평가

❖ 한 성 호 ❖ shan@postech.edu
Virginia Polytechnic Institute & State University
산업시스템공학과 박사
현 재: 포항공과대학교 산업공학과 교수
관심분야: 휴먼-컴퓨터 인터페이스, 감성공학, 제품디자인

❖ 권 민 정 ❖ mj77.kwon@samsung.com
포항공과대학교 산업공학과 석사
현 재: 삼성전자 정보통신 총괄 근무
관심분야: 인체공학적 제품 설계, UI, Universal design

❖ 이 승 희 ❖ iemodel@tmax.co.kr
포항공과대학교 산업공학과 석사
현 재: 티맥스소프트 선임연구원
관심분야: Bioinformatics

논 문 접 수 일 (Date Received) : 2004년 10월 26일
논 문 수 정 일 (Date Revised) : 2004년 03월 27일
논문게재승인일 (Date Accepted) : 2006년 04월 26일