

이용자 인터페이스 설계 원칙과 평가방법*

User Interface Design Principles and Evaluation Methods

이 지 연**

Jee Yeon Lee

차 례

- | | |
|-------------------|-------------------|
| 1. 서 론 | 4. 이용자 인터페이스 평가방법 |
| 2. 인간-컴퓨터 상호작용 | 5. 결 론 |
| 3. 이용자 인터페이스 설계원칙 | • 참고문헌 |

초 록

본 연구는 이용자 인터페이스 설계 원칙과 평가방법에 관하여 고찰하고 관련된 사례를 검토하였다. 제품 개발에 중요한 이용자 인터페이스 설계 원칙으로 1) 이용자와 작업 중심의 설계, 2) 기능성 중심의 설계, 3) 이용자 관점에서 설계, 4) 이용자가 작업 수행을 간단, 명료하게 진행하도록 설계, 5) 배우기 쉬운 인터페이스 설계, 6) 단순 데이터가 아닌 정보를 전달하는 인터페이스 설계, 7) 적절한 피드백을 제공하는 인터페이스 설계, 8) 이용자 테스트를 통한 설계 보완 등이 있다. 또한 이 논문은 설계 원칙과 평가방법을 소개하기 위한 기초조사로 인간-컴퓨터 상호작용과 관련된 문제들에 대해서도 기술하였다.

키 워 드

이용자 인터페이스, 인간-컴퓨터 상호작용, 인터페이스 설계, 인터페이스 평가방법

* 이 글은 2002년 5월 10일 정보관리학회 주최로 KISTI에서 열린 HCI주제의 2002 정보관리강좌에서 발표한 내용을 정리한 것임.

** 연세대 학교 문헌정보학과 강사

(Lecturer, Dept. of Library & Information Science, Yonsei University, jlee01@lis.yonsei.ac.kr)

ABSTRACT

This paper investigates the user interface design principles and evaluation methods with the corresponding examples. Some of these principles are: 1) focus on the users and their tasks, not the technology; 2) consider function first, presentation later; 3) conform to the users' view of the task; 4) do not complicate the user's task; 5) promote learning; 6) deliver information, not just data; 7) design for responsiveness; and 8) try it out on users, then fix it. In addition, this paper deals with the general human computer interaction issues as the basis for introducing the user interface design principles.

KEYWORDS

User Interface, Human Computer Interaction, Interface Design, Interface Evaluation

1. 서론

인간-컴퓨터 상호작용은“ 이용자와 상호작용을 하는 컴퓨터 시스템의 설계, 구현 및 평가를 연구하는 분야”로 정의할 수 있다(ACM 1992). 이 분야에서 연구하는 영역은 사용자-기계간의 협력 작업, 사용자-기계간 커뮤니케이션 구조, 설계분야와 사회 및 기관과의 상호작용, 학습성, 인터페이스 알고리즘과 프로그래밍, 공학적 측면의 인터페이스 구축, 상세화과정, 인터페이스 설계와 구현, 설계의 절충적인 요소 등이다(Dix et al. 1998).

서투른 설계는 크고 작은 실수를 유발하는데, 이러한 크고 작은 부주의에 대처할 수 있는 안정되고 일관성 있는 설계를 하는 것은 쉬운 일이 아니다. 사용자 인터페이스는 시스템 구축에 있어 마지막 단계에 끼워 넣어지는 부분이 아니라 전체 시스템의 모든 부분과 통합되어 시스템 설계와

구현의 전 단계에 걸쳐 처리되어야 하는 부분이다. 사용자 인터페이스 설계는 보기 좋은 화면의 구성을 의미하는 것이 아니고 이용자의 요구를 받아들여 요청 작업을 수행하는 과정을 지원해야 하며 이용자의 부주의한 실수에 유연하게 대처할 수 있어야 한다.

Shneiderman(1998)은 사용자 인터페이스 평가의 중요부분으로 다섯 가지의 측정 가능한 인적 요소를 지적하고 있는데 이 요소들은 다음과 같다.

- (1) 배우는데 걸리는 시간: 이용자가 작업 수행에 적합한 명령어나 기능을 배우기 위한 시간이 얼마나 필요한가?
- (2) 작업 실행 속도: 벤치마크 작업을 수행하는데 시간이 얼마나 걸리는가?
- (3) 사용자 실수율: 벤치마크 작업을 수행하는데 이용자는 얼마나 많은, 그

리고 어떤 종류의 실수를 범하는가?

- (4) 기억력: 이용자는 한 번 이용한 시스템의 이용법을 얼마나 오랫동안 기억할 것인가? 기억력은 배우는데 걸린 시간, 이용 빈도 등과 관련이 있다.
- (5) 이용자의 주관적인 만족도: 시스템의 다양한 기능들을 이용하는 것에 대한 이용자의 선호도는 얼마나 되는가? 이 질문에 대한 대답은 인터뷰나 설문조사 등을 통하여 얻어질 수 있다.

모든 인터페이스 설계자들이 위의 다섯 가지 범주에 대하여 모두 성공적인 인터페이스를 목표로 삼지만 많은 경우 이들은 절충요소(tradeoffs)들에 직면하게 된다. 이러한 절충요소에 대해서는 이 논문의 전체에 걸쳐 적절한 부분에서 언급되어 있으며 결론부분에서 요약, 정리되어 있다.

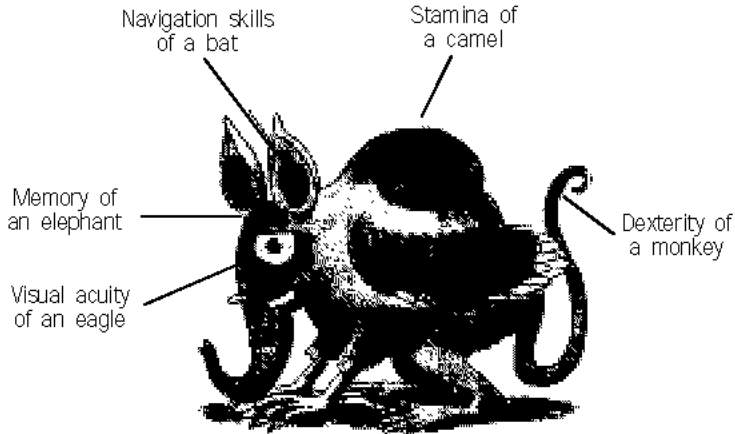
2. 인간-컴퓨터 상호작용

인간-컴퓨터 상호작용에서 일컫는 '이용자'는 개인이용자, 협력 작업을 하는 이용자 그룹, 또는 한 조직 내의 이용자들을 의미한다. 각각의 이용자는 수행하고자 하는 작업을 가지고 있으며 이용자는 기술적인 요소를 이용하여 자신이 원하는 작업을 실행시키고자 하는 사람을 말한다. 또한 '컴퓨터'는 일반적인 데스크톱 컴퓨터에서부터 큰 규모의 시스템까지 통칭하는 말이다. 그리고 시스템은 '비컴퓨터적인' 요소, 즉 다른 이용자도 포함하는 개념이다.' 상

호작용'의 의미는 이용자와 컴퓨터 사이의 직접적 혹은 간접적인 일체의 커뮤니케이션을 말한다. 직접적 커뮤니케이션은 작업 실행 과정에서 일어나는 명령과 피드백을 포함하는 의사 전달 과정을 지칭하며 간접적 커뮤니케이션은 작업의 후면 처리, 혹은 일괄 처리를 의미한다. 인간-컴퓨터 상호작용을 정리한다면, 이용자가 자신이 원하는 작업을 수행하도록 컴퓨터에게 지시를 내리고 컴퓨터의 반응을 인식하여 대응해 나가는 행위라 할 수 있다.

인간-컴퓨터 상호작용의 원칙을 이용자 인터페이스 설계에 적용함에 있어 인터페이스 설계자들은 많은 노력을 기울여 바람직한 결과물을 얻으려 한다. 그러나 더 나은 인터페이스를 위한 고정된 원칙은 존재하지 않는다. Dix 외 (1998)는 인터페이스 설계 기술을 가치로 표현한다면 그 가치의 90%는 무정형의, 그리고 이용자를 대상으로 하여 유도된 것이라 하고 있다. 비록 인터페이스 설계자 자신이 이용하기 쉽고 이해하기 쉬운 인터페이스를 만들었다 믿을 지라도 일단 이용자로 하여금 초기 단계에 이용해 보도록 하는 것은 아주 가치 있는 과정이다. 이용자들이 설계과정에 참여하여 설계의 초기단계에 문제점을 파악하여 미리 수정하는 것은 차후의 변화를 최소화함으로써 프로그래밍 작업에 있어 시간과 비용 면에서의 경제성을 가져오게 할 수 있다.

<그림 1>은 이용자 인터페이스 설계 시에 설계자들이 상상해 볼 수 있는 이용자의 이미지이다. 종종 범하는 인터페이스 설계의 실수 중 하나는 완벽한 이용자를 대



<그림 1> 완벽한 이용자(The Perfect User, SIGCHI Bulletin, 2002)

상으로 하는 인터페이스를 만들려고 하는 것이다. 그러나 현실의 이용자는 인터페이스 설계자들이 상상하는 이용자들보다 훨씬 덜 완벽한 경우가 대부분이다. 이용자의 능력은 성별, 신체별, 지적인 면에서 개인차가 있고, 심지어 같은 이용자도 심신의 건강상태나 피로도에 따라 능력차가 나타날 수 있다.

컴퓨터는 이용자 이외에 인간-컴퓨터 상호작용의 또 하나의 필수적인 부분이다. 컴퓨터의 용량과 접근방식은 인터페이스 설계에 있어 중요한 제한요소가 될 수 있다. 상호작용에서 컴퓨터에 연관된 제한요소들을 열거한다면, 첫째 컴퓨터가 기능을 수행하는데 속도와 관련된 것으로 빠른 속도의 컴퓨터를 이용하면 이용자가 심리적 부담감 없이 상호작용에 임하게 된다. 둘째는 저장에 관련된 문제로 디스크에서 메모리로 데이터를 전달할 때 병목현상 등이

생길 수 있음을 의미하는데 이 역시 시스템 속도에 영향을 미치게 된다. 셋째로 그래픽 정보와 관련된 것으로 그래픽을 담고 있는 인터페이스의 경우 컴퓨터가 얼마나 빠르게 디스플레이를 하는가와 관련이 있다. 마지막으로 네트워크 용량은 자원이나 파일 공유를 허용하지만 작업실행면에서 효율성이 떨어지는 결과를 가져올 수 있다.

상호작용은 '이용자와 시스템 간에 이루어지는 커뮤니케이션'이라 정의할 수 있다. Norman(1988)은 상호작용모델로 실행-평가 모델(Execution-evaluation cycle)을 제시하였는데 이 모델은 상호작용을 하는 시스템의 목적을 '이용자가 자신의 목적을 수행하도록 지원하는 것'이라고 규정하고 있다. 실행단계는 4 단계로 이루어져 있는데, 1) 목표수립, 2) 의도형성, 3) 행위의 순차적 상세화, 4) 행위의 실행이 그것이다. 다음으로 평가단계는 5) 결과의 인지,

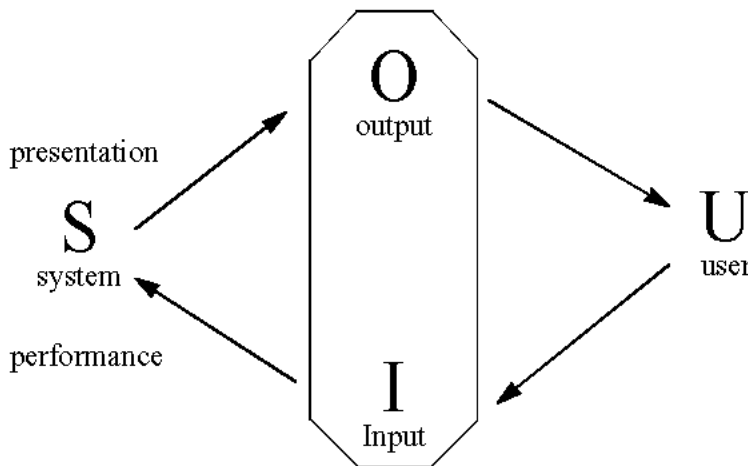
6) 결과의 해석, 7) 목표, 의도에 근거하여 결과를 평가하는 세 단계로 구성된다. Norman의 모델은 계획을 수립하고 인터페이스 상에서 이를 실행시킨 후 이용자는 인터페이스에 나타난 실행 결과를 보고 평가를 한 후 그 다음 단계의 행위를 결정한다는 것이다.

Norman(1988)은 또한 ‘ 상호작용 ’에서 인터페이스가 이용자에게 문제점을 야기시키는가에 대해 기술하고 있다. 그는 여기에서 ‘ 실행상의 격차(Gulfs of execution)’와 ‘ 평가상의 격차(Gulfs of evaluation)’를 언급하고 있다. 실행상의 격차란 이용자가 행하고자 하는 행위와 시스템이 허용하는 행위에 차이가 있는 경우를 말하며, 평가 상의 격차란 이용자가 기대하는 결과가 실제 시스템에 의해 생성되고 보여지는 결과가 다른 경우를 지칭한다.

<그림 2>는 또 다른 상호작용의 틀을 보

여주고 있다. 이 모델은 상호작용에 있어 시스템을 더 분명하게 드러냄으로써 보다 현실적인 상호작용의 과정을 나타내고 있다. 입력과 출력으로 표현되는 부분이 인터페이스를 형성하는 부분이고 이용자와 시스템을 연결하는 매개체 역할을 한다. 이때, 이용자와 시스템은 각각 고유의 언어를 사용하므로 ‘ 상호작용 ’이 서로 다른 언어의 해석을 담당하게 된다. 상호작용에서의 문제는 해석이 어렵거나 불가능한 경우에 발생하게 된다. 이용자의 의도는 인터페이스에 의해서 해석되어 시스템으로 전달된다. 이 의도는 시스템 상태로 전환되어 처리되어지고 그 결과가 인터페이스에 의해 디스플레이 되고 이용자에 의해 해석, 평가되어진다.

다음은 이용자 인터페이스에서 채택되어 사용되는 상호작용의 유형으로 <표 1>은 이러한 유형들을 정리하여 보여주고 있다.



<그림 2> 상호작용 틀(Interaction Framework, Dix et al, 1998)

3. 이용자 인터페이스 설계 원칙

이용자에 대한 이해와 이용자의 요구사항을 반영하는 인터페이스 설계의 중요성이 이해되고 확산된 반면, 인터페이스 설계자들이 이러한 이용자 중심의 인터페이스를 설계하는데 지침으로 삼을 원칙 수립에 대한 연구 및 노력은 상대적으로 미약한 실정이다. 이용자 인터페이스 설계의 원칙 제시에 대한 노력은 Norman(1988)에서 그 기원을 찾을 수 있으며 Dix 외(1998)나 Johnson(2000)과 같은 인간-컴퓨터 상호작

용(HCI) 분야의 연구자들이 나름대로의 인터페이스 설계 원칙을 제시하고 있다. 이들의 주장은 모두 이용자를 이해해야 한다는 것에서 설계 원칙의 제시를 시작하고 있으며 이 부분에 대하여 특히 강조하고 있다. 본 논문에서 다룬 이용자 인터페이스 설계 원칙 부분은 이 세 연구자의 주장을 중심으로 살펴본다.

3.1 Norman의 설계원칙

Norman(1988)은 이용자가 이해하기 쉬

<표 1> 상호작용 유형

상호작용 유형	설 명
명령어방식	컴퓨터에 명령문을 직접 입력 장점 : 유연성, 반복적 작업에 유용하게 이용 단점 : 명령어 암기, 작업지시에 도움이 되는 정보가 없음, 전문성을 갖춘 이용자에게 적합
메뉴방식	기억(Recall) 보다 인식(Recognition)에 의존하는 방식 메뉴 아이템의 집합과 명칭이 이용자에게 유용한 암시를 충분히 제공하여야 함
자연언어이용 방식	자연 언어의 모호성이 혼동을 가져올 수 있음 이용 가능한 영역이 제한되어 있음
질의/응답 및 대화 방식	질문과 응답으로 구성된 단계별 상호작용 배우고 이용하기 쉬우나 기능상 제한이 따름
스프레드시트	데이터 입력과 검색에 효과적으로 이용 데이터 값을 자유로이 조작할 수 있으나 유연성 있는 데이터 입력과 출력 요구됨
윈프(WIMP)	윈도우를 이용하는 시스템 Windows, Icons, Menus, Pointers를 말하기도 하고 Windows, Icons, Mice, Pull-down menus를 지칭하기도 함 아이콘은 임의로 정한 기호이므로 이용자가 이해하기 어려운 경우가 있음
포인트와 클릭	하이퍼텍스트 개념과 밀접한 관련이 있음 터치스크린 환경에서 사용될 수 있음
삼차원 공간 이용	공간적인 여유를 제공하고 자연스러운 입체감을 연출할 수 있음

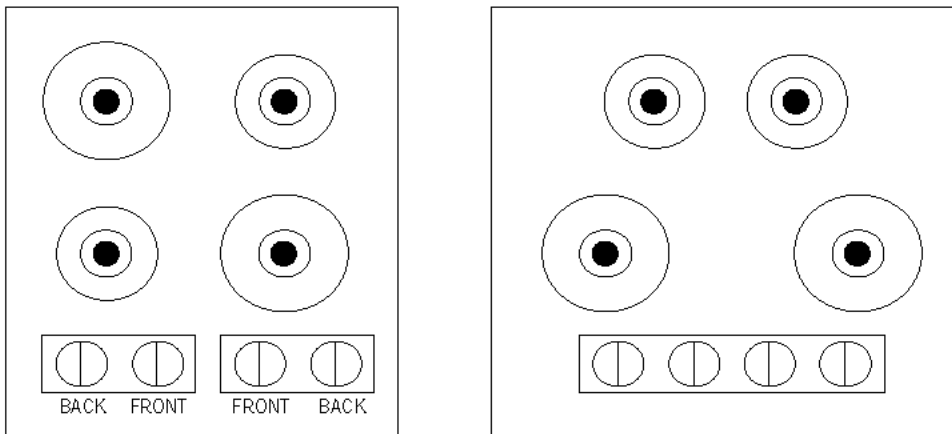
은 인터페이스를 컴퓨터에 국한하지 않고 일상적인 생활용품 전반에 대해 다루고 있다. 문의 손잡이나 오븐 등 이용자가 흔히 접하는 물건에도 이용자를 위한 설계가 중요하다고 강조한다. Norman은 우리가 늘 이용하는 제품의 설계가 그 제품의 효용성을 결정하는 중요한 요소임을 강조하고 있으며 자신의 의견을 응용심리학적 관점에서 풀어나가고 있다. 이용자가 목표를 수립하여 자신의 의도를 형성하고 행위를 상세화 하여 실행시킨 후 결과를 인지, 해석하고 평가하는 단계를 거친다는 실행-평가 모델(Execution-evaluation cycle)과 각각의 단계에서 발생하는 문제들을 실행-평가 격차(Execution-evaluation gulfs)로 규정하며 이러한 격차를 최소화하기 위해서 고려해야 하는 사항들로 다음의 네 가지를 들고 있다.

첫째는 가시성(Visibility)이다. 가시성은

제품의 중요한 부분은 눈에 띄어야 하고 그런 부분들의 의미를 바르게 전달할 수 있어야 한다는 것이다. 예를 들어 텔레비전의 경우 이용자가 많이 사용하는 파워, 채널전환, 음량조절의 기능은 TV와 리모콘에서 눈에 잘 띄도록 배치하여야 하며 각 기능의 의미를 잘 전달하는 표현이나 기호를 사용해야 한다.

둘째는 대응(Mapping)의 원칙이다. 대응은 어떤 기능을 통제하는 조절장치와 그 기능을 담당하는 부분이 잘 연결되어 표현되는 것을 의미한다. <그림 3>은 스토브의 가열부분과 스위치부분의 배열을 보여주고 있는데 왼쪽은 부분적 대응이 이루어진 예이고 오른쪽은 대응이 자연스러운 경우이다.

셋째는 행동유도성(Affordance)으로, 물건들은 각각 모양이나 다른 특성에 의해 그것들을 어떻게 이용하는가에 대한 암시를 제공한다는 것이다. 예를 들어 버튼이나



<그림 3> 부분 대응과 완전한 대응¹⁾

1) Donald A. Norman의 *The Design of Everyday Things*, pp.76-77에 있는 예를 정리하여 도안한 것임

부저는 눌러야 한다는 행위를 지시하고 문의 손잡이도 모양에 따라 밀기를 유도할 수 있기도 하고 잡아당기는 방향으로 설계될 수도 있다. 전체가 툴유리로 된 문의 경우 출입구가 명확하지 않은 경우가 있는데 이 때 출입구 아래에 깔개를 놓으면 사람들로 하여금 어느 곳으로 드나들어야 하는지 지시를 하는 기능을 한다.

Norman이 강조하는 설계에 있어서 중요한 부분의 하나는 적절한 피드백의 제공이다. 적절한 피드백은 사용자들의 작업수행에 있어 동기부여에 중요한 요소이며, 만약 시스템의 작업수행에 지연이 발생했을 때 사용자가 좌절이나 포기를 하지 않도록 하는데 있어 피드백은 중요한 기능을 발휘한다(Norman 1988).

3.2 Dix 외의 설계원칙

Dix 외(1998)는 이용성이 있는 시스템을 설계하기 위하여 인간-컴퓨터 상호작용에 대한 이해가 필요하다고 지적하고 있다. 또한 인터페이스는 사용자가 의도하는 작업을 지원하고 이용자의 부주의한 실수를 포용할 수 있어야 한다고 주장하고 있다. 모든 실수는 서투른 설계에서 기인한다고 보고 있으며 인터페이스 설계자들이 고려해야 할 사항으로 다음을 들고 있다.

- 이용자의 작업에 적합해야 한다.
- 이용하기 쉬워야 하고 이용자의 지식이나 경험 수준에 따라 이용자에 맞는 기능이나 내용이 제공되어야 한다.
- 작업실행에 대한 피드백이 제공되어야 한다.

- 정보의 디스플레이가 이용자에게 적당한 형식과 속도로 이루어져야 한다.
- 인간공학적 측면을 고려해야 한다.

위의 사항들에 대한 구체적인 뒷받침으로 이용자 모델링, 작업분석, 유용성 테스트, 설계과정 등에 관련된 여러 가지 방법들이 적용될 수 있다. 이용자 요구 모델링, 인지 모델 등을 통해 이용자의 요구나 인식, 정신작용의 측면들을 파악할 수 있고 이용자의 물리적, 사회적, 문화적 환경에 대한 이해가 인터페이스 설계에 유용하다고 하겠다. 또한 이용자가 수행하는 작업을 이해하고 작업수행을 지원하는 인터페이스 설계를 위하여 작업분해(Task decomposition), 지식기반분석(Knowledge-based analysis), 개체-관계분석(Entity-relationship-based analysis)이 이용되고 있다. 그리고 인간공학적인 입장에서 인터페이스 설계를 포함하는 인간-컴퓨터 상호작용 연구에 있어 인간의 신체적 특징의 이해를 바탕으로 해야 한다고 주장한다. 인간공학과 관련된 사항으로 기능조작을 담당하는 요소들의 배치는 기능성, 작업의 순서, 이용 빈도를 고려해야 하며, 시스템을 이용하는 장소, 이용자 집단의 특성, 이용시의 자세 등 물리적 환경에 대한 이해가 필요하고 아울러 이용자 건강, 색채의 이용도 고려해야 한다.

Dix 외는 상호작용의 이론과 원칙적인 면을 일반적 입장에서 기술하고 있는데 각각의 특수한 입장과 환경적 여건을 고려하여 적절한 방법을 채택하고 응용하는 것은 인터페이스 설계자들의 역할이라 하겠다.

3.3 Johnson의 설계원칙

Johnson(2000)은 사용자 인터페이스 블루퍼(bloopers)에 대한 저서에서 자신의 책은 초보 사용자 인터페이스 설계자들을 위한 지침서라고 소개하고 있다. Johnson은 특정 설계 규칙을 제시하기보다는 일반적인 그래픽 사용자 인터페이스 설계 규칙의 기본이 되는 원칙을 제시하고 있는데, Johnson의 설계 원칙은 <표 2>에서처럼 여덟 가지로 요약될 수 있다.

① 첫 번째이자 가장 중요한 원칙으로 이용자와 이용자가 수행하는 작업을 중심으로 하는 설계를 강조하고 있다. 이를 위해서 다음의 질문에 대한 답을 구할 것을 권장하고 있는데, 이러한 질문들은 “누가 주된 사용자인가? 이용자에게 어떤 기능을 제공하는 시스템을 설계하는가? 현재 이용자의 작업 수행 방식의 장점과 단점은 무엇인가? 이용자의 기술과 지식 정도는? 이용자가 선호하는 작업 수행 방식은 무엇

인가?” 등을 포함한다. 이용자를 이해하고 이용자의 특성을 파악하며 이용자와 협력하는 것이 인터페이스 설계의 핵심적인 부분이라 할 것이다

② 두 번째 원칙은 인터페이스 설계에 있어서 기능적인 면이 보여지는 것에 우선해야 한다는 점을 강조하고 있다. 좋은 사용자 인터페이스는 “화려한 화면”이 아니라 시스템이 제공하는 기능 혹은 역할에 염두에 두는 인터페이스를 의미한다. 즉 인터페이스의 목적, 구조, 조직 등을 고려해야 함을 지시하는 것이다.

③ 사용자 관점에서의 설계는 몇 가지 측면에서 설명될 수 있는데, 우선 사용자로 하여금 부자연스러운 기능 수행을 요구하지 말아야 한다. 예를 들어 체스게임을 하는 소프트웨어에서, “이동(Move)” 명령어 사용을 위해 명령어 모드를 선택해야 한다던가, 여러 개의 반복적인 명령을 구별하기 위하여 이용자가 각각의 명령에 이름을 부여해야 한다던가 하는 요구사항을 제거해야 함을 의미한다. 또한 특정문자를 이용하

<표 2> Johnson의 사용자 인터페이스 설계 원칙

	설계 원칙
원칙 1	이용자와 작업 중심의 설계
원칙 2	기능성 중심의 설계
원칙 3	이용자 관점에서의 설계
원칙 4	이용자가 작업 수행을 간단, 명료하게 진행하도록 설계
원칙 5	배우기 쉬운 인터페이스 설계
원칙 6	데이터가 아닌 정보를 전달하는 인터페이스 설계
원칙 7	적절한 피드백을 제공하는 인터페이스 설계
원칙 8	이용자 테스트를 통한 설계 보완

고 문자수를 제한하는 파일 이름을 사용하게 하는 등의 임의적인 제한요소들을 제거해야 한다. 다음으로는 언어사용에 있어 이용자들이 이해하기 쉬운 단어나 표현을 사용하며 프로그램 코딩에 관련된 전문용어가 이용자에게 노출되지 않도록 해야 한다.

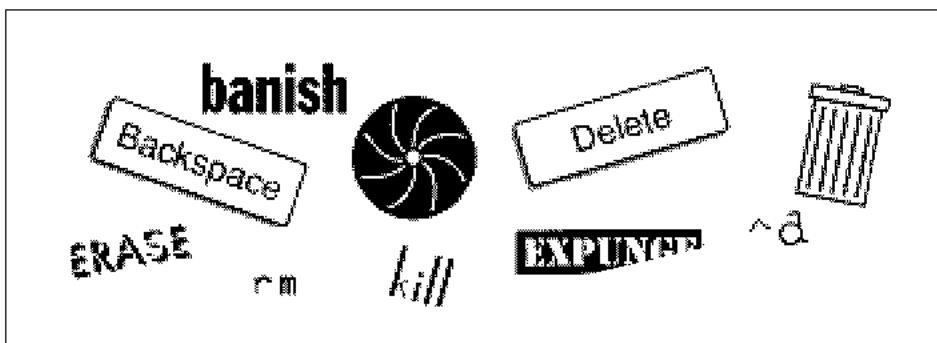
④ 다음으로 인터페이스는 이용자가 작업을 간단하게 처리하도록 지원해야 하는데 이용자에게 요구되는 노력의 범위를 최소화시켜야 함을 의미한다. 예를 들어, 자주 사용하는 기능의 경우 초기화를 시킨다든가, 마법사(wizard)를 이용하고, 이용자별 요구사항을 반영시키는 커스터마이제이션이 이루어져야 한다. 또한 이용자가 기술적인 측면을 이해할 필요가 없도록 해야 한다.

⑤ 배우기 쉬운 인터페이스의 설계와 관련하여 두 가지 다른 접근법을 생각할 수 있는데 안에서부터 밖으로의 사고(Think inside-out)와 밖에서 안으로의 사고(Think outside-in)가 그것이다. 이용자 인터페이스 설계자들이 흔히 범하는 실수가 첫 번째의 방식을 적용하는 것인데 이

는 시스템과 기술적인 면을 이용자에게 요구하여 인터페이스에 반영하는 것이다. 바람직한 인터페이스는 이용자가 이해하고 알고 있는 방식을 시스템에 적용시키는 것이다. 또한 배우기 쉬운 인터페이스는 일관성 있는 설계를 통하여도 이루어진다. 간단한 예로 <그림 4>는 삭제와 관련된 명령어로서 여러 가지가 있음을 보여준다. 인터페이스에서 이용되는 기능이나 단어들의 일관성을 유지하면 이용자들이 새로운 시스템을 배우는 데 도움이 될 것이다.

⑥ 단순 데이터가 아닌 정보를 전달하는 인터페이스는 이용자로 하여금 중요한 정보를 쉽게 볼 수 있도록 하고 인터페이스 설계 시에 보는 시점의 이동순서, 전달되는 정보에 적합한 매체를 사용하고 상세 부분에도 주목할 수 있도록 적절한 배치를 하여야 한다.

⑦ Norman과 마찬가지로 Johnson도 피드백의 중요성을 강조하고 있는데 이는 시스템이 이용자에게 나타내는 반응은 작업 실행의 성공여부와는 별개의 것임을 의미한다. 이용자가 작업실행을 명령했을 경우



<그림 4> 시스템에서 다르게 이용되는 삭제 명령어들

시스템은 명령의 이행을 수행하는 것과 병행하여 이용자로 하여금 명령을 인식하였음과 현재 처리중임을 표시하여야 한다. 또한 Johnson은 실행시간이 0.1초 이상인 경우는 커서의 모양을 변환하여야 하며 1초 이상인 경우는 남은 실행 시간을 표시할 것을 권장하고 있다. 이는 0.1초가 넘으면 인간이 기대하는 원인과 결과에 대한 예상치가 깨지기 때문이며, 1초의 개념은 인간의 의사소통에서 나온 것으로 1초 이상의 침묵은 인간이 불편한 상태를 느끼도록 한다.

⑧ 마지막으로 이용자 테스트가 인터페이스 설계에 중요한 요소임을 강조하고 있다. 이용자 테스트를 통하여 인터페이스의 문제점을 파악하도록 하고 이를 시스템 개발자에게 인식시켜서 수정, 보완하도록 한다.

Johnson이 제시하는 이용자 인터페이스 설계의 원칙은 이용자라는 하나의 중심체로 모아지고 있다. 이용자와 이용자 요구에 대한 이해를 바탕으로 하는 인터페이스 설계를 최우선의 원칙으로 삼고 있다. 이러한 중심 개념의 세부사항으로 일관성 있는 설계, 배우기 쉬운 인터페이스, 적절한 반응의 제공 등을 제시하고 있다.

4. 인터페이스 평가방법

이용자 인터페이스에 대한 평가는 설계의 전 과정을 통하여 이루어지는 총체적인 부분으로 시스템 구축 전, 구축이 진행되는 동안, 그리고 구축 후로 나누어 시행될 수

있다. 구축 전단계의 평가는 문제점을 일찍 파악하여 수정을 위한 비용을 최소화시킨다는 장점이 있다. 구축 전단계에서 사용되는 대부분의 기술들은 분석을 위주로 하며 전문가가 참여하여 설계를 평가한다. 이전에 다른 곳에서 행해졌던 평가의 결과나 모델링 방법이 이 단계에서 이용되기도 한다. 일단 프로토타입이나 실제 시스템이 만들어지면 실험적 혹은 관찰적 방법을 이용하는 평가기술이 사용된다. 질의 방식을 이용하여 이용자들이 개인적인 견해를 유도하기도 한다.

인터페이스 평가의 목적은 설계의 기능성과 유용성을 테스트하여 문제점을 파악하는 것이다. 이러한 인터페이스 평가는 실험실 평가와 이용자 작업장 평가로 나누어 시행된다.

평가방법으로는 이용자가 업무를 수행하는 일련의 과정을 단계적으로 분석하여 평가하는 인지적 진행평가(Cognitive Walkthrough), 시스템 평가를 위한 원칙과 지침이 되는 리스트(Heuristics)를 만들어서 전문가가 평가를 하는 휴리스틱 평가(Heuristic evaluation), 실험의 결과를 가지고 평가를 하는 방법(Review-based evaluation), 특정 설계 모델의 세부사항을 가지고 평가하는 모델 기반의 평가 방법(Model-based evaluation) 등이 있다. 이러한 평가방법은 업무에 적합한 것으로 조심스럽게 선택되어야 하는데 평가방법 선정에 고려할 사항으로 평가가 이루어지는 단계, 평가스타일, 평가방법의 주관성과 객관성, 제공된 기타 정보, 평가 데이터 획득의 즉각성, 평가에 필요한 자원 등이

있다.

위에서 평가방법의 유형과 선정에 대해 이론적이고 개념적인 기술을 시도한 반면, 이후의 부분은 인터페이스 평가에 대해 실질적 접근을 시도하고자 한다. 인터페이스 설계를 평가할 때 고려할 사항을 크게 네 부분으로 나누어서, 개선해야 할 인터페이스적 요소에 대한 평가를 내리고자 한다. 우선 그래픽 이용자 인터페이스의 설계 평가와 관련된 네 가지 측면을 열거한다면 다음과 같다.

1) 인터페이스 구성요소 : 인터페이스를 구성하는 요소들로 버튼, 메뉴, 다이얼로그 박스 등 여러 가지가 있는데 이런 구성요소들의 적절한 선택과 올바른 사용이 인터페이스를 평가하는 부분이 된다.

2) 구성요소의 배치와 외양 : 적절하게 선택, 사용된 구성요소들을 인터페이스에 어떻게 배치하여 보여주는가도 좋은 설계의 필수요소이다.

3) 인터페이스를 통한 사용자-시스템 상호작용 : 사용자-컴퓨터 상호작용과 관련된 인터페이스 설계는 사용자 인터페이스 설계 원칙과 밀접한 관련이 있다. 이는 단순히 구성요소의 단편적인 면을 평가하는 것이 아니라 사용자가 인터페이스 이용시에 느끼게 되는 만족감 등과 관련이 있다.

4) 반응 : Norman과 Johnson이 강조한 피드백 개념의 연장으로 반응은 사용자가 작업을 수행하는 속도를 얼마나 잘 지원하는가, 아니면 간섭적인 요소로 작용하는가에 관한 것이다.

위의 네 가지 요소를 인터페이스 설계자들이 부주의하게 이용하는 경우 많은 설계상의 실수가 발생하는 데 각각의 요소에 대하여 좀 더 자세히 살펴보고자 한다.

4.1 GUI 구성요소와 인터페이스 평가

GUI 개발 도구는 사용자 인터페이스 작성에 필요한 다양한 구성요소를 지원하는 것으로 많은 사용자 인터페이스들이 이런 개발 도구를 적절히 사용하지 못하는 경우를 발견할 수 있다. 인터페이스 설계자들의 실수를 유형별로 본다면, 작업영역이나 이용자에 대한 이해의 결여에서 발생하는 구성요소 이용의 의미상의 실수가 있고 (Hackos & Redish 1998) 또 다른 유형은 프로그래머들이 GUI 개발 도구를 적절히 사용하지 못하는 데서 발생한다. 적절하지 못한 GUI 개발 도구를 선정하여 인터페이스를 구축하는 것도 구성요소 측면의 실수를 유발하게 만든다. GUI 구성요소를 잘못 사용하는 경우는 이용자들로 하여금 인터페이스가 조잡하고 전문성이 없다는 느낌을 갖도록 한다. 또한 GUI 구성요소와 관련된 문제들은 이용자가 인터페이스를 어렵다고 느끼도록 할 것이다.

GUI 구성요소에 관련된 설계 실수는 인터페이스 기능을 의도한 대로 접근하는 데 있어 복잡한 방해요소로 작용할 수 있다. 예를 들어, '다이나믹 메뉴'의 경우 인터페이스 공간 활용을 위해 임의적으로 사용할 때 이용자들이 혼란을 경험할 수 있고 자신이 원하는 기능이 숨어있을 경우 초보 이용자가 찾지 못하는 경우도 발생할 수

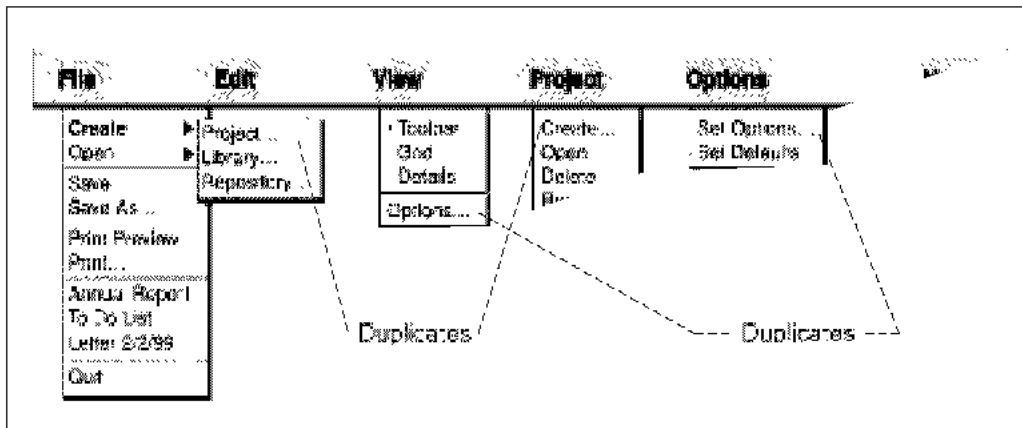
있다. 또 다른 예는 ‘중복된 메뉴 아이템’인데 하나의 명령어를 두 곳 이상의 메뉴 바에 위치시키면 이용자는 상당히 혼란을 느낄 것이다. 더 심각한 실수는 같은 명령어를 서로 다른 이름으로 여러 군데에 위치시키는 것이다. <그림 5>에서 보여주는 화면은 중복된 메뉴 아이템을 이용하는 실수를 보여주고 있다.

또 다른 GUI 구성요소에 해당하는 설계 실수는 ‘숨은 기능’의 사용이다. 숨은 기능이란 인터페이스 구성요소를 통하여 직접 접근할 수 있는 것이 아니라 키보드 조합이나, 문맥적 상황에서 표시되는 팝업 메뉴에 의해 접근이 가능한 기능으로 이용자는 이러한 기능에 대한 존재를 인터페이스를 통하여 알 수 없게 된다. 이는 가시성을 고려한 설계에 저촉되는 예가 될 수 있다. 이 외에 그래픽 기능에 상응하는 단축키의 결여도 문제가 될 수 있는데 이 경우는 특히 전문 수준의 이용자에 있어 유용성을 절감시키는 영향을 미치게 된다.

4.2 구성요소의 배치 및 외양 인터페이스 평가

구성요소의 배치는 각각의 구성요소가 인터페이스 상에서 어떻게 표현되고 서로 어떤 관계를 나타내는가에 대한 문제이다. 한 가지 주의하여야 할 점은 배치나 나열의 아주 사소한 실수라 하더라도 심각한 유용성 문제를 야기할 수 있다는 것이다. 예를 들어 다이얼로그 기능 통제 버튼과 내용 통제 버튼을 같이 붙여서 나열하는 방법이다. 각기 다른 기능을 담당하는 통제 요소들은 분리되어 배치되어야 한다. <그림 6>은 ArcSoft의 Photo Studio의 화면으로 통제 버튼들이 화면 상단에 혼합되어 나열되었다.

구성요소의 배치와 관련하여 배치의 순서도 중요한 평가기준이 될 수 있다. 이용자의 작업 수행 순서에 적합한 배치를 하여야 하는데 이와 관련하여 1) 대부분의 이용자가 읽어가는 순서, 2) 이용 빈도, 3) 다른 기능 버튼과의 관련성, 4) 이용자가



<그림 5> 중복된 메뉴 아이템을 포함한 인터페이스 (Johnson, 2000)

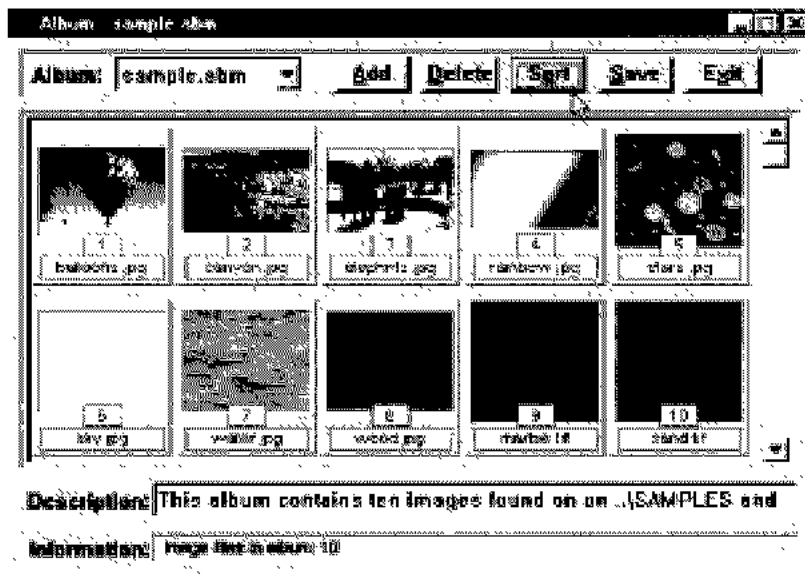
기대하는 방식 등을 고려해야 한다. 다른 기준으로는 그룹박스의 적절한 이용, 올바른 레이블링과 스페이스 사용, 글자 폰트와 크기 등이 있다.

4.3 상호작용 인터페이스 평가

상호작용과 관련된 평가는 시스템의 다이나믹한 측면을 다루는 것으로 시스템이 어떻게 표현되는가 보다는 어떻게 기능을 수행하는가에 초점을 맞추고 있다. 상호작용과 관련된 인터페이스의 문제는 숫자적으로는 다른 범주의 실수에 비해 적지만 이용자에게 미치는 영향은 더 크다. 왜냐하면, 상호작용 범주의 실수는 범위가 포괄적이어서 하나의 실수를 수정하기 위해 여러 가지 구성요소의 사용이나 배치 등 다른 범주의 실수를 수정해야 하는 경우가

생기기 때문이다. 그리고 상호작용의 실수는 인터페이스에 직접적으로 나타나지 않기 때문에 원인을 찾기가 쉽지 않다. 상호작용 실수의 예는 시스템 구현의 제한을 이용자와 상호작용에 노출시키는 경우인데 시스템이 이용자에게 인터페이스 사용이 있어서 임의적인 제한을 부여하는 경우이다. 예를 들어, 스페이스를 포함하는 파일명을 사용할 수 없다던가, 로그인 아이디를 8자로 제한한다던가 하는 등의 시스템 제약이 이용자에게 요구되는 경우이다.

이용자가 기대하는 방식의 화면 연결도 상호작용에서 중요한 역할을 담당한다. <그림 7>은 화면의 연결에 일관성이 없어서 이용자로 하여금 혼란을 겪고 방향성을 잃도록 만들 가능성이 있는 예로 Altavista의 웹페이지 연결을 보여주고 있다. 왼쪽 화면의 경우 Altavista의 홈페이지에서 'People



<그림 6> 기능 버튼 혼용의 예

Finder'를 선택한 경우인데 연결이 Home Email & Tools People Finder로 되어서 선택하지 않은 Email & Tools 부분이 중간에 이미 거친 단계로 표시되어 있다. 오른쪽 화면은 왼쪽 화면에서 'Email & Tools'를 선택하여 나온 결과인데 결과가 하나 앞의 단계로 Home Tools이고 명칭도 정확히 일치하지 않는다. 이런 식으로 화면의 연결이나 명령의 표현이 정확하지 않은 경우 이용자는 작업수행의 어려움을 겪게 된다.

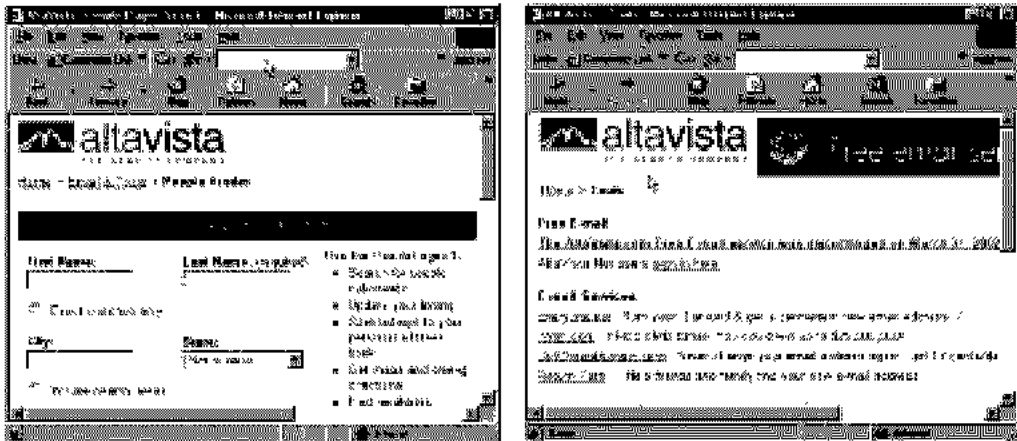
4.4 적절한 반응 제공 인터페이스 평가

앞에서 언급했듯이, 반응은 실행이나 속도를 의미하는 것이 아니다. 실행 면에서 낮은 평가를 받는다 하더라도 반응 면에서 높은 평가를 받는 시스템이 있을 수 있다. 반응은 이용자의 요구에 대하여 인터페이스가 시스템과 연결하여 얼마나 즉각적이고 충분한 대응을 하는가와 관련된 것이다. 문자키 입력상태, 포인터의 변환, 마우스

클릭 등 이용자의 모든 행위에 인터페이스는 즉각적인 반응을 나타내어야 한다. 또한 명령 실행이 지연될 경우 커서의 모양 전환 등을 통해서 적절한 정보를 이용자에게 나타내 주어야 한다. 시스템이 대기 상태에서 다음 예상 명령이 있는 경우 사전처리를 시작해서 명령이 들어왔을 경우 바로 필요한 본 처리과정만 거쳐 이용자에게 결과를 전달하는 경우도 있다. 시스템은 또한 이용자가 하나의 명령에 대한 결과를 전달하기 전에 또 다른 명령을 한 경우, 순서대로 처리하여 결과를 출력하기보다는 실행 속도가 빠른 명령을 먼저 처리하여 보여주는 방법을 이용하는 경우도 있다.

5. 결 론

현실적으로 이용성의 절충요인에서 벗어나서 이용자의 요구만을 수렴할 수 있는 인터페이스를 설계하는 것은 어려운 일이



<그림 7> 상호작용과 관련된 인터페이스 설계 실수의 예

다. 예를 들어, 이용자에게 인터페이스 이용법을 배우는 데 오랜 시간을 제공할 경우, 작업 실행 속도는 세련된 작업요청 명령방식의 사용으로 훨씬 빨라질 수 있다. 또 이용자 실수율을 최소화하는 인터페이스를 위해서 작업 실행 속도는 희생되어야 하는 경우도 있을 수 있다. 몇몇 경우는 결과에 대한 이용자의 개인적이고 주관적인 만족도가 성공적인 인터페이스와 시스템의 결정요소가 될 수 있다. 또 다른 경우는 시스템 이용방법을 빨리 배우고 빠른 작업 실행 속도를 제공하는 설계가 성공의 결정요인이 될 수 있다.

이런 주관적이고 상호절충의 요소가 인터페이스 설계의 평가에 관여될 수 있지만 궁극적으로 인터페이스 설계자들이 목표로 삼고 있는 것은 유용성의 극대화이다. 유용성과 관련하여 두 가지 질문을 할 수 있는데, 그 하나는 '유용성을 보장하는 인터페이스 설계 방법은 무엇인가?' 하는 것이고 다른 하나는 '인터페이스의 유용성은 어떻게 포착되고 측정될 수 있는가?'에 대한 것이다.

이 질문들에 대한 대답을 구할 수 있는 한 가지 방법은 인간-컴퓨터 상호작용의 연구가 이루어진 여러 분야의 지식을 바탕으로 하여 효과적인 상호작용에 대한 구체적인 원칙을 유도하여 수립하는 것이다. 인간심리, 컴퓨터, 사회현상 관련 연구 분야의 지식을 기반으로 한 원칙들은 인터페이스 설계나 평가에 방향성을 제시할 수 있을 것이다.

유용성을 보장하는 인터페이스 설계의 지침으로 이용자 참여 설계(participatory

design) 방법을 들 수 있다. 이는 이용자가 설계의 과정에 개입되어 자신들의 의견을 수렴하는 설계가 이루어지도록 하는 방법이다. 본문에서 언급한 바와 같이, 인터페이스 설계의 가장 최우선 원칙은 이용자와 이용자의 요구사항에 대한 이해이다. 그리고 이용자의 요구사항을 가장 잘 반영할 수 있는 방법은 이용자를 자신들이 이용할 인터페이스 설계 작업에 참여시키는 것이다. 이용자를 설계과정에 참여시키는 방법이 비용이나 시간적인 면에서 비능률적일 수 있으나, 인터페이스 완성 후에 사용자들의 평가에 의해 발견되는 문제점들을 다시 보완해야 하는 점을 고려한다면 결과적으로는 비용이나 시간적으로 오히려 능률적인 방법일 수 있다.

현재 구현되는 많은 상호작용 시스템들은 과거에 비해 유용성이 우수한 것으로 평가되고 있다. 이 점으로 미루어 보건대, 앞으로 개발되는 시스템은 상호작용면에서 현재보다 더 우수한 인터페이스를 제공할 것으로 예상된다. 컴퓨터 기술의 발전은 더 나은 인터페이스를 위한 하드웨어를 제공할 것이며 인간과 컴퓨터의 커뮤니케이션 폭을 확대시킬 것이다. 그러나 기술적인 발달이 유용성을 향상시키는 유일한 요소가 될 수는 없고 이용자의 특성을 고려한 이용자 중심의 시스템 개발이 이루어질 때 기술발전은 더욱 빛을 보게 될 것이다. 인터페이스 설계 원칙들은 기술적인 면보다는 이용자 쪽에 더 의존적이며 인간-컴퓨터 상호작용에 있어 인간적인 요소, 즉 이용자에 대한 폭 넓고 깊은 이해에 바탕을 두고 있다고 하겠다.

참고문헌

- ACM Special Interest Group on Computer-Human Interaction Bulletin. 2002. *ACM SIGCHI Bulletin March/April 2002 Issue*
- ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group. 1992. *ACM SIGCHI curricula for human-computer interaction*. Technical report, New York: ACM.
- Dix, A., Finlay, J. Abowd, G., & Beale, R. 1998. *Human-Computer Interaction* 2nd Ed. London: Prentice Hall
- Hackos, J.T. & Redish, J.C. 1988. *User and Task Analysis for Interface Design*. New York: John Wiley & Sons
- Johnson, J. 2000. *GUI Bloopers: Don't's and Do's for Software Developers and Web Designers* New York: Morgan Kaufmann.
- Norman, D.A. 1988. *The Psychology of Everyday Things* Basic Books, New York
- Shneiderman, B. 1996. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* 3rd Ed. Addison Wesley Longman.