

□ 특별기고 □

공중 데이터베이스 서비스를 위한 메뉴 방식 사용자 인터페이스의 설계

서울대학교 컴퓨터공학과 김혁만* · 이석호**

● 목	차 ●
I. 서 론 II. 화면 설계를 위한 지침 2.1 화면 밀도 2.2 시각적 그룹 2.3 Highlighting 2.4 공간 배치 2.5 텍스트 디스플레이	III. 메뉴 설계를 위한 지침 3.1 메뉴의 구조 3.2 메뉴항목의 수와 메뉴의 깊이 3.3 메뉴항목의 이름과 분류 3.4 메뉴항목의 순서와 선택 3.5 메뉴의 탐색 IV. 결 론

I. 서 론

1970년대초 영국 우정성의 연구팀에 의해 첫선을 보이게 된 비디오텍스타는 대중매체는 이미 전세계 많은 국가에서 보급되어 널리 이용되고 있다. 선진국에서는 이미 오래전부터 컴퓨터 기술과 통신 기술이 결합된 새로운 데이터통신 기술을 이용하여, 필요한 각종 정보를 컴퓨터의 기억장치에 축적시켜 두었다가 필요한 때 단말기를 통해 온라인으로 찾아보는 시스템이 보편화되어 있다. 특히 영국의 PRESTEL, 프랑스의 TELETEL, 일본의 CAPTAIN 등은 시스템의 성능, 제공되는 정보의 질, 그리고 점책적 배려에 힘입어 성공적으로 운영되고 있다. 우리나라에서도 80년대 중반부터 데이터베이스 서비스 분야에 대한 집중적인 연구 및 투자가 이루어져 한국경제신문사의 KETEL, DACOM의 천리안 II, 한국통신의 KTAINFO 등과 같은 상용 시스템들이 다양한 분야의 정보를 일반 사용자들에게 제공하고 있다.

그러나 이러한 공중통신망을 이용한 데이터베이스 서비스가 더욱 대중화되기 위해서는 정보제공자가 제공하는 정보가 신속 정확해야 하며, 시스템의 응답속도가 빨라야 하고, 정보검색 방법이 간단하여 교육을 받지 않은 일반인도 용이하게 사용할 수 있어야 한다. 또한 사용자들이 정보를 검색하고 검색된 정보를 받아보는, 사용자와 시스템 간의 정보의 중계자 역할을 수행하는 시스템의 사용자 인터페이스는 사용방법이 간단하고 효율적인 정보 검색 수단을 제공하여야 하며, 장시간 정보를 검색하여도 가능한한 사용자의 검색효율을 떨어뜨리지 않는 형태의 화면을 제공하여야 한다. 즉 사용자 인터페이스는 변경할 수 있는 설계의 대상이지만 그것을 사용하는 인간은 주어진 조건이므로, 사용자 인터페이스는 인간의 정보처리 기능에 최대한 조화되도록 설계하여야 한다. 따라서 사용자 인터페이스는 인간공학적인 측면을 최대한으로 고려한 화면을 통하여 정보를 디스플레이하여야 한다.

한편 공중 데이터베이스 서비스를 제공하는 인터페이스들은 주로 메뉴방식을 사용하고 있다. 메뉴방

* 정회원
** 중신회원

식의 사용자 인터페이스는 화면 해석시간 및 메뉴 탐색시간, 메뉴항목 선택 어려움을 최소화하면서 사용자 친숙도를 극대화할 수 있도록 설계해야 한다. 이를 위하여는 인간공학적 측면을 고려한 설계 지침을 먼저 정의한 후 이에 따라 최상의 인터페이스가 될 수 있도록 시스템을 설계하고 구현하여야 한다.

또한 어떤 설계 지침도 없이 구현한 기존의 사용자 인터페이스도 좋은 설계지침에 따라 재설계하여 다시 구현하였을 경우 많은 성능향상을 이룰 수 있다는 연구결과들이 있다. [19]에 의하면 기존의 화면을 인간공학적 측면을 고려한 화면 설계 기준에 맞추어 재설계하였을 경우 화면 해석시간이 약 40% 정도 단축되었고, [11]에서는 기존의 메뉴를 인간공학적 메뉴 설계기준에 맞추어 재설계하였을 경우 메뉴항목 선택 어려가 약 40% 정도 단축되었다. 즉 화면과 메뉴에 대한 설계 지침을 정의하고 이에 따라 사용자 인터페이스를 (재)설계하면 보다 효율적인 시스템의 구현이 가능하며, 화면과 메뉴에 대한 설계 지침은 인간공학적인 측면을 최대한 고려하여 사용자의 입장에서 가장 편하고 효율적으로 화면을 통하여 메뉴 항목을 선택할 수 있도록 정의하여야 한다. 본 고에서는 공중 데이터베이스 서비스를 위한 메뉴방식 사용자 인터페이스의 화면과 메뉴 설계를 위한 인간공학적인 측면을 고려한 설계 지침 및 이에 따른 고려사항들을 논한다. 본 고에서 제시하는 설계 지침들은 일반적인 소프트웨어의 사용자 인터페이스에도 적용이 가능하다.

II. 화면 설계를 위한 지침

사용자 인터페이스에서 제공하는 화면은 사용자의 심리적, 인간공학적 측면 그리고 사용자와 설계자의 경험을 바탕으로 설계하는 것이 바람직하며, 좋은 화면을 설계하기 위하여는 화면 밀도, 시각적 그룹, highlighting, 공간 배치, 텍스트 디스플레이 등 여러 요소들을 고려하여 설계하여야 한다.

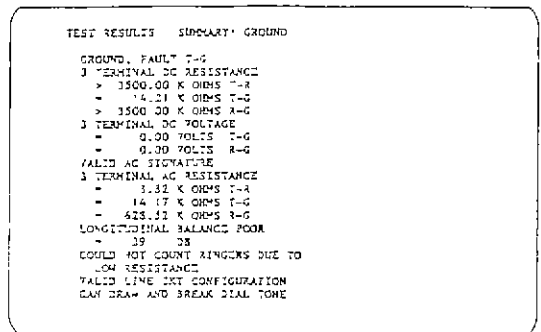
2.1 화면 밀도

화면 밀도(screen density)란 한 화면에서 정보 표시 가능 영역에 대하여 현재 정보가 나타나 있는 영역의 비율로서 다음과 같은 여러 지침들이 제시되고 있다. Danchak은 화면 밀도는 25%를 넘지 않아야 하며 15%가 최적이라고 주장하였고, Smith는 24×80

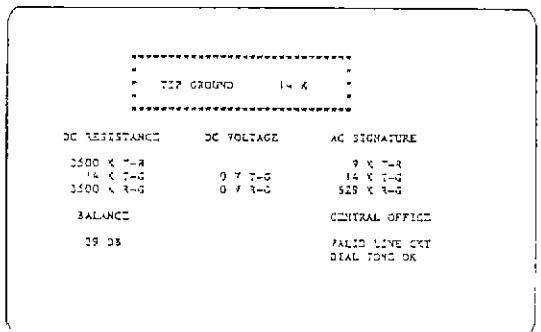
텍스트 모드 CRT의 경우 표시되는 문자의 수(화면 밀도)에 따라 화면을 복잡(600 문자, 화면 밀도 31.2% 이상), 보통(300~600 문자), 단순(300 문자, 화면 밀도 15.6% 이하)한 화면의 세 종류로 구분하여 복잡한 화면은 피해야 한다고 하였으며, NASA에서는 60%를 화면 밀도의 최대값으로 제시하였다[20]. 이들이 주장하는 화면 밀도의 값은 상당한 차이가 있지만 모두 화면 밀도의 최대값을 제시하여 이를 넘지않는 화면, 즉 화면의 전체 영역중 일부분만을 사용하는 화면을 구성하도록 권하고 있다.

한편 어떤 화면들은 화면 밀도는 거의 비슷하나 전혀 다르게 보일 수 있다. 예를 들어 화면의 한쪽 부분에 정보가 몰려서 나타나는 경우와 화면 전체에 넓게 흩어져 있는 경우에는 화면 밀도가 같을지라도 여백의 존재 형태가 화면을 전혀 다르게 보일 수 있다. 이럴 경우에는 일반적으로 화면에서 정보가 나타나는 부분들을 3~5개의 여백으로 구분하여 시각적으로 구별할 수 있도록 하는 것이 바람직하다.

Tullis[18]는 전화선에 대한 실험 결과를 디스플레이하는 두개의 화면(그림 1)을 이용하여 사용자의



(a) narrative format 화면



(b) structured format 화면

(그림 1) 화면 밀도에 관한 실험

화면 해석 시간을 분석한 결과, narrative format으로 표현된 화면 보다 같은 정보를 나타내지만 화면 밀도를 축소시킨 structured format 화면의 화면 해석 시간이 약 40% 단축됨을 알게 되었다.

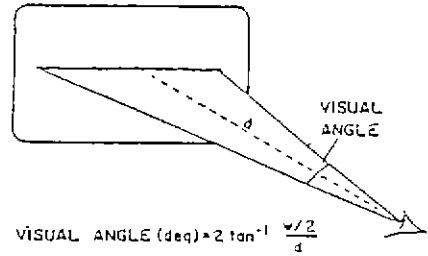
화면 밀도를 축소하기 위하여 약어를 사용하는 경우가 있으나 이때에는 모든 사용자가 충분히 약어에 익숙한 계층이어야 하며, 시스템은 약어 사전을 제공하여야 한다. 또한 화면 설계자는 디스플레이할 정보의 내용과 사용자들의 정보 요구사항을 정확히 분석하여 사용자가 원하는 정도의 정보만을 화면에 나타내야 한다. 예를 들어 사용자는 소수점 이하 한 자리수까지 만을 보기 원하나 시스템은 소수점 이하 세 자리수까지 계산하였다고 하여 전부 디스플레이할 필요는 없다. 한편 열 이름을 이용한 표를 적절히 사용하면 중복을 최소화하면서 정보를 보기 쉽게 표현할 수 있다.

2.2 시각적 그룹

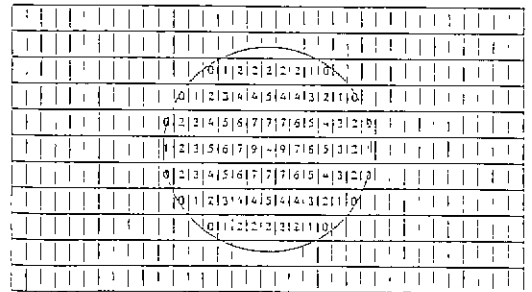
비슷한 형태 또는 내용의 정보를 시각적으로 그룹(group)화시켜 디스플레이하면 읽기 쉬우므로 사용자가 정보를 포착하기 편하고, 그룹별로 정보를 해석하기 쉬우며, 정보 그룹 간의 관계를 쉽게 강조할 수 있다.

인간의 시각은 수정체의 구조상 어떤 점을 중심으로 상하 좌우 각각 2.5° (0.044 rad) 만큼의 원에 가장 민감한 반응을 나타낸다. 즉 직경 0.088 rad(5°)인 원, 5-degree visual angle이 가장 눈에 민감한 부분(그림 2 참조)으로 CRT를 475 mm의 거리(최적 거리 450~500 mm의 평균)에서 쳐다본다면, 5° visual angle은 CRT상에 직경 41.8 mm의 원으로 나타난다. 만일 CRT상의 문자 간격이 2.8 mm이고 줄 간격이 5.6 mm라면, 5° visual angle은 (그림 3)과 같이 88개의 문자를 포함하는 영역이 된다. 일반적으로 화면상에 성보를 그룹별로 디스플레이하면 직사각형의 형태로 나타나므로 원으로 표현된 5° visual angle을 사각형으로 표현하면 (12~14문자) X (6~7줄)의 사각형으로 표시가 가능하다[7,8,19].

Tullis[19]는 임의의 문자를 기준으로 그 문자와 수평으로 2 문자 이내, 수직으로 0 문자 만큼 떨어져있는 모든 문자들을 연결한 그래프를 만들 경우 이 그래프에 연결된 모든 문자들의 집합을 그룹이라고 정의하고, 서로 상이한 특성을 갖는 52개의 화면을 화면 분석 프로그램을 이용하여 분석하고 다음과 같은



(그림 2) visual angle



(그림 3) 화면상의 5-degree visual angle

질론에 도달하였다.

- 화면 검색시간은 그룹의 수와 크기에 따라 결정됨

- 큰 그룹을 몇 개의 작은 그룹으로 분할하면 검색시간이 단축되나, 그룹의 수가 어느 밀정한 값을 넘어서게 되면 검색시간은 그룹의 수에 비례하여 증가하게 됨

- 그룹의 크기가 5-degree visual angle 보다 작을 경우 검색시간은 그룹의 수와 비례하고, 그룹의 크기가 5-degree visual angle보다 클 경우 검색시간은 그룹의 크기에 비례함

따라서 화면은 최소의 그룹 수로 이루어지도록 설계하며, 이때 각 그룹의 크기는 가능한한 5-degree visual angle에 가깝도록 설계하여야 한다.

그 외에 그룹별로 다양한 색조를 사용하여 그룹간의 관계나 어느 특정 그룹을 다른 그룹보다 두드러지게 할 수도 있으며, 여백으로 그룹을 분리하기 보다는 그래픽 문자를 이용하여 그룹의 경계선을 그으므로써 그룹간의 차이를 명확히 할 수도 있고, 명암이나 역상 등을 이용하여 그룹을 돋보일 수도 있다.

2.3 Highlighting

highlighting은 화면의 어느 특정 부분에 사용자의

주의를 집중시키기 위한 기법으로써 역상, 명암, flashing, 밑줄, 색상 등의 방법을 사용할 수 있다. 역상은 어떤 특정 그룹을 화면상에서 돋보이게 하는 매우 효율적인 방법이나 역상으로 표시된 그룹에 사용자의 주의를 너무 집중시키므로 역상으로 표시할 그룹은 신중히 결정하여야 한다. 이 방법은 주로 메뉴에서 이미 선택한 메뉴항목을 나타내기 위하여 사용한다.

명암은 화면의 밝기를 여러 단계로 조정하여 텍스트를 디스플레이하는 기법으로서, 너무 많은 단계의 밝기를 한 화면에서 사용하면 사용자의 지각 능력이 이들의 차이를 인식하지 못할 가능성이 있다. 따라서 밝기는 밝고 어두운 두 단계만 사용하는 경향이 있다.

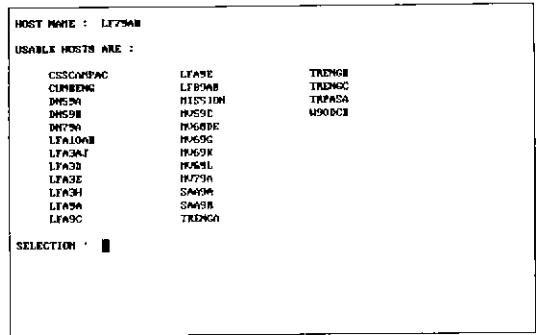
flashing은 화면에서 특정 부분의 디스플레이 방법을 주기적으로 변경하는 것으로 flashing할 부분을 반복해서 켜다 끄는 경우, 주기적으로 명암의 값을 변경시키는 경우, 주기적으로 역상을 반복하는 경우가 있다. flashing은 여러 highlighting 기법 중 사용자의 주의를 가장 집중시킬 수 있는 방법으로 프로그램이나 데이터에 심각한 이상이 있을 때 또는 에러가 발생했을 때 주로 사용한다.

시스템이 밑줄문자나 문자 밑에 밑줄을 그을 수 있는 기능을 제공한다면 특정 단어나 문장에 밑줄을 사용할 수 있다. 만일 밑줄문자나 문자 밑에 밑줄을 그을 수 있는 기능을 시스템이 제공하지 못한다면 dash(-)를 밑줄그을 문자 바로 밑에 사용할 수도 있다. 밑줄은 주로 테이블에서 열 이름을 강조하기 위하여 사용한다. 또한 문자나 화면 배경에 색조를 사용하므로써 특정 부분을 강조할 수도 있다.

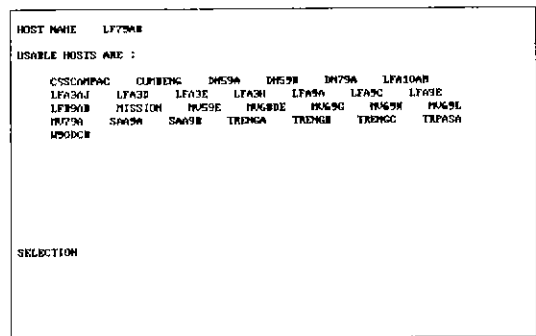
한편 여러가지 highlighting 기법을 화면에 적절히 사용하면 화면 설계자가 원하는 방향으로 사용자의 주의를 집중시킬 수 있으나, 한 화면에 너무 많은 highlighting 기법을 적용할 경우에는 오히려 사용자의 주위가 분산되는 crossword-puzzle 효과[6]가 발생하므로 highlighting 기법의 사용에는 세심한 주의가 필요하다.

2.4 공간 배치

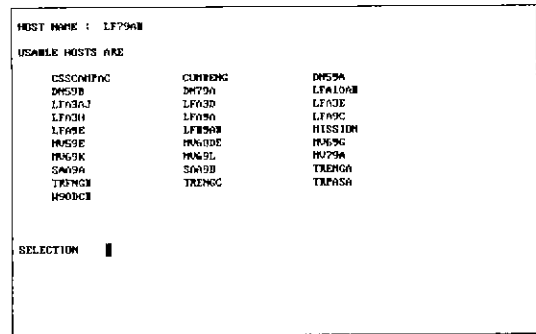
화면상에 정보를 디스플레이할 때에는 관련된 정보들을 어떻게 배치해야 사용자가 빠르고 편하게 정보를 인식할 수 있는가를 고려하여야 한다. Wolf의 실험[20]에 의하면 서로 연관성이 있는 여러 데이터들의 리스트를 화면에 디스플레이할 때에는 알파벳의 순서로 수평으로 일정한 간격을 두고 나열하는 방법



(a) 열단위로 디스플레이한 화면



(b) 줄단위로 디스플레이한 화면



(c) 출순서에 따라 열단위로 디스플레이한 화면

(그림 4) Wolf의 실험

보다는 수직 방향으로 열을 맞추어 디스플레이하는 것이 효율적이라고 하였다. 그는 어떤 네트워크에서 액세스 가능한 호스트 컴퓨터들의 이름을 디스플레이하는 5종류의 상이한 화면을 이용하여 화면 해석 시간을 비교하였다. (그림 4)는 실험에 사용한 화면의 일부로 (a)는 호스트 컴퓨터 이름을 알파벳 순서대로 열단위로 디스플레이한 화면이고, (b)는 알파벳 순서대로 수평으로 일정한 간격을 두고 줄단위로 디스플레이한 화면으로서, 실험 결과에 의하면 (a)의 화면 해석시간이 (b)보다 37% 감소하였다.

한편 최악의 화면 해석시간을 갖는 화면은 (그림 4)의 (c)와 같이 수직으로 열을 맞추어 데이터들을 디스플레이하나, 그 순서는 수평의 줄에 맞추어 나열된 경우이다. (c)의 화면 해석시간이 가장 낮은 이유는 사용자들이 화면을 열단위의 그룹으로 나누어 인식하는 경향이 있으므로 데이터들의 나열 순서도 열단위 그룹 내의 순서일 것이라고 가정하게 되나, 실제로 데이터들은 줄순서에 따라 나타나기 때문이다. 이와 같은 형태의 화면은 실제로 많이 사용되고 있으나 Wolf의 실험결과에 따르면 데이터 리스트는 줄단위 보다는 열단위로, 나열순서는 열순서에 맞추는 방법이 가장 효과적이다.

열단위로 데이터를 나열할 경우에는 데이터들을 좌측 혹은 우측으로 정렬시킴으로써 화면의 해석을 쉽게 할 수 있다. 일반적으로 문자 데이터는 좌측 정렬, 수치 데이터는 우측 정렬을 원칙으로 하며 소수점을 포함하는 수치 데이터는 소수점에 맞추어서 정렬시켜야 한다. 또한 데이터들 사이에 계층적인 관계나 포함관계가 존재할 경우에는 indentation을 사용하면 효과적이다.

데이터를 그 내용을 서술하는 라벨과 함께 디스플레이하면 화면 해석이 더욱 쉬워질 수 있다. 라벨과 데이터를 같이 디스플레이할 경우에 라벨은 그들끼리 좌측 정렬시켜 디스플레이하는 방법이 일반적이다. 그러나 라벨을 좌측정렬시킬 경우 라벨과 데이터 사이에 많은 여백이 삽입되어 그들간의 연관성이 잘 표현되지 않을 수가 있으므로 다음과 같은 방법을 사용할 수 있다.

- movie-credit style: 라벨들을 우측 정렬시킴[6]
- 라벨과 데이터를 선으로 연결
- 라벨들을 좌측 및 우측으로 정렬시킴(한글에서만 사용 가능)

2.5 텍스트 디스플레이

화면에 텍스트를 디스플레이 해야 할 경우에는 다음의 몇 가지 사항을 고려하여야 한다. 일반적으로 영문 텍스트를 디스플레이할 경우에는 대문자와 소문자를 함께 사용하는 것이 효율적이다. Poulton 등의 실험[14]에 의하면 대문자와 소문자를 함께 사용한 텍스트가 대문자만을 사용한 텍스트 보다 읽는 속도가 약 13% 빠르다고 하며, Vartabedian의 실험[21]에 의하면 화면에서 대문자만으로 이루어진 특정 단어를 찾는 속도가 소문자만으로 이루어진 단어를 찾는

것보다 약 13% 빠르다고 한다.

텍스트 상에 나오는 단어들 사이의 간격두기 방법에는 균일 간격두기와 불균일 간격두기가 있다. 균일 간격두기는 모든 단어들 사이의 간격이 균일(일반적으로 하나의 여백)하도록 디스플레이하는 것으로서, 이 방법은 주로 좌측정렬과 같이 사용하므로 텍스트의 좌측선은 일정하나 우측은 들쭉날쭉하게 된다. 반면에 불균일 간격두기는 단어들 사이의 간격이 일정하지 않은 것으로서, 이 방법은 좌측 및 우측정렬을 같이 적용하여 텍스트를 디스플레이할 때 텍스트의 좌우측선을 일정하게 유지하며 한 단어가 분리되어 두 줄에 이어지지 않도록 여백을 삽입하여 단어들 간의 간격을 조절하는 방법이다. Trollip 등[17]은 특정 텍스트를 워드프로세서를 이용하여 좌우측 정렬을 이용한 불균일 간격의 텍스트와 좌측만 정렬시킨 균일 간격의 텍스트로 만들어 두 그룹의 실험대상자들에게 실험한 결과 균일 간격 텍스트가 불균일 간격 텍스트보다 읽는 속도가 약 11% 정도 빠르게 나타났다. 즉 균일 간격 텍스트가 불균일 간격 텍스트보다 화면상에서 보기에는 못하지만 사용자가 읽는 속도면에서는 효율적임을 알 수 있다.

또한 여러 줄의 텍스트를 디스플레이할 때 줄 사이의 간격은 double space가 single space 보다 11% 정도 더 빠르며[9], 텍스트의 절들은 최소한 한 개 이상의 빈 줄로 시각적으로 구분하므로써 하나의 절이 하나의 그룹을 이루게 하는 것이 좋다[6].

일반적으로 사용하는 80문자 CRT에서 텍스트를 디스플레이할 때 한 줄에 몇 개의 문자를 표시하는가에 대한 문제에는 일치된 견해는 없다. Tullis는 한 줄은 40~60 문자로 이루어져야한다고 주장하였으며[19], Bouma는 한 줄에는 60 문자 이상의 문자가 있어서는 안된다고 하였다[3]. Duchnicky 등은 78 문자와 52문자의 줄로 구성된 텍스트의 읽는 속도를 비교한 결과 큰 차이가 없으나, 두 경우 모든 26 문자의 줄로 이루어진 텍스트 보다는 현저히 빠르게 읽을 수 있음을 실험을 통하여 확인하였다[14]. Gould 등은 visual angle이 16~36° 사이에서는 텍스트를 읽는 속도가 거의 비슷하며 이 각도 범위를 벗어나는 경우에는 읽는 속도가 현저히 저하됨을 밝혔다. 보통 visual angle이 16°이면 CRT 상에서 약 50 문자, 36°이면 약 130문자를 나타낼 수 있으므로 줄의 길이가 약 50~130 사이에서는 CRT상의 텍스트를 읽는 속도가 거의 차이가 없으나, 그 범위를 벗어나는 줄 길이를 갖는 텍스트를 읽는 속도는 현

저히 떨어지게 됨을 알 수 있다[7].

III. 메뉴 설계를 위한 지침

인간공학적 측면을 고려하지 않고 만들어진 사용자 인터페이스는 사용자에게 보다 큰 정신적 부담을 주며 잦은 오류를 유발하여 전체적인 효율을 저하시킨다. 따라서 사용자 인터페이스는 사용법을 쉽고 빠르게 익히고 오랫동안 기억할 수 있어야 하며, 사용자의 오류를 줄이고, 사용속도를 향상시킬 수 있도록 인적 요인을 고려하여 설계하여야 한다.

일반적으로 메뉴방식의 사용자 인터페이스는 명령어의 형식을 정확히 알 필요없이 단순히 메뉴항목을 이해하고 선택만 하면 되므로 일반인을 대상으로 하는 공중 데이터베이스 서비스에는 메뉴 방식이 널리 사용되고 있다. 그러나 메뉴방식은 다단계의 메뉴항목을 디스플레이하기 위하여 많은 화면이 필요하므로 메뉴 선택과 메뉴 디스플레이로 인한 시간 소모가 크다. 따라서 메뉴방식의 사용자 인터페이스 설계시에는 메뉴의 구조, 메뉴항목의 수와 메뉴의 깊이, 메뉴항목의 이름과 분류, 메뉴항목의 순서와 선택방법, 그리고 메뉴의 탐색방법에 대한 깊은 고려가 필요하다.

3.1 메뉴의 구조

메뉴방식이란 화면상에서 디스플레이되는 메뉴항목들의 집합인 메뉴에서 사용자가 하나의 메뉴항목을 선택하면 화면의 내용이나 상태가 변하는 시스템으로, 메뉴 시스템에서는 제공하는 수 많은 메뉴항목들을 분류하고 그들간의 포함 관계를 고려하여 각각의 메뉴를 설계한다. 이때 상위메뉴는 하위메뉴를 개념적으로 포함하여야 하며, 하위메뉴로 갈수록 그 내용이 구체적으로 서술된다.

메뉴 시스템은 트리 구조, 사이클이 없는 네트워크 구조, 사이클이 있는 네트워크 구조의 형태로 구성이 가능하다[15]. 트리 구조는 일반적으로 사용하는 메뉴의 구조이며, 어떤 메뉴항목에 대한 탐색 경로를 여러개 만들고자 할 경우에는 트리 구조 대신 네트워크 구조를 사용하고, 특히 사이클이 있는 네트워크 구조는 메뉴항목들의 복잡한 개념적 관계를 서로 연관시키기 편하다. 네트워크 구조를 사용할 경우에는 메뉴 탐색중 사용자가 현재 있는 위치를 잃어버릴 가능성이 있으므로 여러가지 다양한 탐색 명령어가 필요하다.

3.2 메뉴항목의 수와 메뉴의 깊이

CRT 화면에 디스플레이할 각각의 메뉴 화면을 구성할 때에는 디스플레이할 화면의 크기를 고려하여 메뉴 화면을 설계하며, 이때 crowding 방식이나 funneling 방식으로 설계가 가능하다. crowding 방식은 화면크기가 허용하는 한 메뉴항목들을 한 메뉴 화면에 설계하고, 만일 화면이 넘치게 되면 새로운 메뉴 화면을 설계하는 방법이다. 예를 들어 64개의 메뉴항목을 하나의 메뉴로 만든다면 사용자는 64개의 메뉴항목을 모두 검사하고 서로를 비교하여 원하는 한 항목을 선택하므로 사용자의 메뉴항목 선택시간은 매우 많이 필요하다, 시스템은 디스플레이할 메뉴의 수가 적으므로 부하가 감소된다[16].

funneling 방식은 한 메뉴에서 메뉴항목의 수를 최소로 하여 사용자가 쉽게 원하는 메뉴항목을 선택하도록 하는 방식으로 전체 메뉴 구조의 깊이가 커지는 단점이 있다. 위의 메뉴를 재구성하여 각 메뉴의 메뉴항목 수를 2로 하면 메뉴의 깊이가 6인 다단계 메뉴 구조를 얻게 되어, 사용자는 모든 메뉴항목을 검색하지 않고 단지 6번의 비교(12개의 메뉴항목 검색)를 통하여 원하는 메뉴항목을 선택할 수 있다. 따라서 사용자의 메뉴 선택시간은 매우 빨라지나, 시스템은 메뉴를 6번 디스플레이해야 하므로 시스템의 부담이 커지게 된다.

funneling을 적용하여 메뉴 구조의 깊이를 크게 할 경우에는 많은 메뉴를 디스플레이하고 그에 대한 응답을 처리하기 위한 시스템의 프로세싱이 많아지게 된다. 즉 하나의 메뉴가 증가할 때마다 시스템은 이 메뉴를 화면에 디스플레이하고 사용자의 선택을 처리하며 이에 대하여 시스템이 어떤 형태로든 응답을 해야한다. 반면에 crowding을 적용하여 한 메뉴에 많은 메뉴항목을 나열하면 사용자는 여러 메뉴항목의 선택에 어려움을 겪게되어 선택시간이 길어져 결국에는 메뉴 시스템에 대한 친밀도가 떨어지게 된다. 따라서 메뉴 구조를 설계할 때에는 각 메뉴에서의 메뉴항목 수와 전체 메뉴의 깊이를 상호 고려하여 사용자가 사용하기 쉬우며 동시에 시스템의 부담도 덜어주는 절충선을 택하여야 한다.

Lee와 MacGregor[10]는 탐색의 형태를 모든 메뉴항목을 읽어본 후 원하는 항목을 선택하는 exhaustive 탐색과 메뉴항목의 탐색중 가장 적당하다고 판단되는 항목을 발견하자 마자 선택하는 self-terminating 탐색으로 구분하여 사용자의 탐색시간을 최소화

하는 최적 메뉴항목의 수를 계산하는 수학적 모델을 제시하였다. b 는 메뉴당 평균 메뉴항목의 수, d 는 메뉴 구조의 평균 깊이, n 은 전체 데이터베이스의 수라면 $\ln n = d \cdot (\ln b)$ 이 성립하며, 한 메뉴당 탐색해야 하는 평균 메뉴항목의 수 $E(b)$ 는 다음과 같다.

$$E(b) = b \quad : \text{exhaustive 탐색}$$

$$E(b) = (b+1)/2 \quad : \text{self-terminating 탐색}$$

또한 하나의 메뉴를 읽고 메뉴항목을 선택하는데 걸리는 시간 S 는 하나의 메뉴항목을 읽는데 소용되는 평균시간 t , 원하는 메뉴항목을 선택하여 해당 키를 누르는 시간 k , 시스템의 응답시간 c 에 의하여 결정된다고 가정하면 S 는 다음과 같이 계산될 수 있다.

$$S = E(b) \cdot t + k + c$$

따라서 원하는 정보를 얻기 위하여 전체 메뉴구조를 탐색하는 시간 ST 는 다음과 같다.

$$ST = d \cdot S = d \cdot (E(b) \cdot t + k + c)$$

$$= \frac{(E(b) \cdot t + k + c)}{(\ln b)} (\ln n)$$

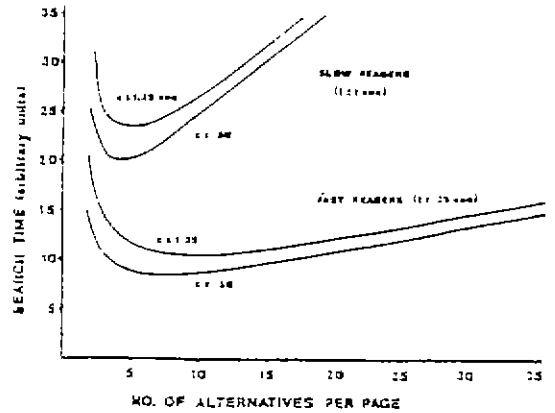
위의 식에서 t, k, c 는 주어진 상수라 가정하고, ST 를 최소로 하는 b 의 값을 구하기 위하여 ST 를 b 에 대하여 미분하여 $d(ST)/db = 0$ 를 풀면 다음의 두 식을 구할 수 있다

$$b \cdot (\ln b - 1) = (k+c)/t \quad : \text{exhaustive}$$

$$b \cdot (\ln b - 1) = 1 + 2(k+c)/t \quad : \text{self-terminating}$$

따라서 위의 두 식에 대하여 t, k, c 를 특정한 값으로 치환하여 풀어보면 <표 1> 및 <표 2>와 같은 b 의 값을 얻을 수 있다. 즉 exhaustive 탐색시에는 한 메뉴의 메뉴항목이 3~8일 때, self-terminating 탐색시에는 4~13일 때 탐색시간이 최소가 됨을 알 수 있다. 또한 (그림 5)는 exhaustive 탐색시의 탐색시간과 메뉴항목의 수와의 관계를 나타내는 그래프로서, 한 메뉴의 메뉴항목이 3~8의 범위를 넘으면 탐색시간이 급격히 증가함을 나타내고 있다.

Lee와 MacGregor는 메뉴의 사용자들이 그 시스템에 익숙하지 않은 경우의 탐색시간 함수를 구하였으나, Paap과 Roske-Hofstrand[13]는 사용자들이 그 메뉴에 경험이 있는 경우의 탐색시간을 구하고 이와 같은 환경에서의 최적 메뉴항목의 수를 계산하였다. 사용자들이 이미 사용할 메뉴에 익숙하면 exhaustive



(그림 5) 탐색시간과 메뉴항목의 수와의 관계

보다는 self-terminating 탐색을 주로 사용하게 되므로, self-terminating 탐색시의 한 메뉴당 탐색해야 하는 평균 메뉴항목의 수 $E(b)$ 는 앞서의 식에서 2를 f 로 치환한 형태가 된다. 여기서 f 는 탐색할 메뉴항목들 중 평균적으로 탐색하는 메뉴항목들의 비율을 나타내는 것으로, $f=2$ 이면 메뉴항목 탐색중 원하는 메뉴항목은 평균적으로 1/2만 검색하면 찾을 수 있다는 의미이고, $f=3$ 이면 1/3만 검색하면 찾을 수 있다는 것을 의미한다. 따라서 f 의 값이 클수록 사용자는 해당 시스템의 사용 경험이 풍부함을 나타낸다.

$$E(b) = (b+1)/f$$

위의 식을 이용하여 같은 방법으로 탐색시간 함수 ST 를 구하고, 이를 b 에 대하여 미분시키면 아래의 함수를 얻을 수 있고, 이 식에서 f 의 값을 2~10으로 변화시키면서 b 의 값을 구하면 b 는 13~37의 값을 갖는다. 즉 사용하는 메뉴에 대하여 사용자의 경험이 풍부할 때는 한 화면당 탐색시간을 최소로 하는 메뉴항목의 수가 커진다.

$$b \cdot (\ln b - 1) = 1 + f \cdot (k+c)/t$$

한편 Lee와 MacGregor가 메뉴항목들이 임의의 순서대로 나열될 상황을 가정한 것에 반하여 McDonald[12]는 메뉴항목들을 그 내용에 따라 그룹별로 분류하였을 때에는 exhaustive 보다는 self-terminating 탐색을 주로 사용하게 되므로, 이때의 탐색시간을 구하고 이와 같은 환경에서의 최적 메뉴항목의 수를 계산하였다. b 개의 메뉴항목으로 구성된 메뉴를 g 개의 그룹으로 나눌 경우 일반적으로 $g = \sqrt{b}$ 의 관계가

있다고 가정하면, 그룹들이 임의의 순서로 나열되고 그룹내에서 메뉴항목들도 임의의 순서로 나열될 때 한 메뉴당 탐색하는 평균 메뉴항목의 수 $E(b)$ 는 다음과 같다

$$E(b) = (g+1)/2 + ((b/g)+1)/2$$

같은 방법으로 탐색시간 함수 ST를 구하여 b에 대하여 미분시킨 후 t, k, c의 값을 변화시키면서 b의 값을 구하면, 최소 탐색시간을 갖는 메뉴항목의 수는 16~36까지 가능하며 대로는 78개를 사용할 수도 있다. 즉 메뉴항목을 그룹지어 디스플레이하면 한 화면에 많은 메뉴항목을 담겨되어 메뉴구조의 깊이를 줄일 수 있으므로 전체 메뉴 탐색시간을 상당히 단축시킬 수 있다

$$\sqrt{b} \cdot (\ln \sqrt{b} - 1) = 1 + (k+c)/t$$

이상의 연구 결과에 의하면 사용자의 탐색시간을 최소화할 수 있는 메뉴화면상에 디스플레이 할 메뉴항목의 수는 3~8(exhaustive 탐색) 혹은 4~13(self-terminating 탐색)이 가장 좋으며, 사용자들이 해당 메뉴에 익숙한 경우에는 최소 탐색시간을 갖는 메뉴항목의 수가 많아지므로 전체 메뉴의 수를 줄일 수 있어 결국 메뉴 구조의 깊이를 작게할 수 있다. 또한 메뉴항목들을 디스플레이할 때 그들의 의미에 따라 그룹별로 나누어 디스플레이하면 더욱 효과적이다.

3.3 메뉴항목의 이름과 분류

메뉴는 상위메뉴가 하위메뉴를 개념적으로 포함하여야 하므로 임의의 메뉴항목은 자신의 하위메뉴 메뉴항목들을 모두 포함하도록 분류하여야 하며, 또한 그 이름도 모든 하위 메뉴항목들을 개념적으로 포함할 수 있도록 지어야 한다. 따라서 상위 메뉴항목에서 하위 메뉴항목으로 탐색해 갈수록 그 내용이 포괄적인 범위에서 하나씩 구체적으로 서술되어야 하며, 다른 범주의 메뉴항목을 포함해서는 안된다.

Lee와 Whaien[11]은 캐나다의 비디오텍스 시스템인 Telidon 시스템을 사용하여 메뉴항목의 이름과 분류에 대한 여러가지 실험을 수행하였다. 첫번째 실험에서는 Telidon 시스템을 사용해 본 경험이 없는 10명의 실험대상자를 선정하고, Telidon 시스템에서 900개의 다양한 주제를 갖는 정보를 임의 추출하여 이를 탐색할 수 있는 79개의 메뉴로 이루어진 트로 구조의 메뉴(평균 메뉴 길이=5) 시스템을 구성하였

다. 이 시스템에서 임의로 16개의 정보를 미리 선택하고, 각각에 대한 질문을 만들어 실험대상자들이 해당 질문의 답을 메뉴 시스템을 통하여 검색하도록 하였다. 그 결과 모든 실험대상자가 평균 49.9%의 에러를 범하였고, 에러의 80%가 6개의 메뉴에서 발생하였으며, 에러의 53%가 최상위 2개 레벨에서 발생하였다. 이 실험에서 발생한 모든 에러는 결국 메뉴 시스템이 잘못 설계되어 발생한 것이므로 실험대상자들이 틀린 선택을 하게된 주원인은 다음의 설계가 잘못된 결과이다.

-분류: 정보가 실험대상자들이 생각하는 카테고리 로 분류되지 않았음

-이름: 정보를 분류한 카테고리 와 메뉴항목의 이름이 적당하지 않음

특히 최상위 레벨에서 에러가 많이 발생한 것은 최상위 레벨의 메뉴항목 이름은 매우 포괄적이기 때문에 그들간의 포함범위가 명확하지 않아서 발생한 것이다.

두번째 실험에서는 첫번째 실험대상자들로 하여금 에러가 가장 많이 발생한 6개 메뉴를 재분류하고 메뉴항목의 이름을 다시 만들게하여, Telidon에 경험이 없는 또 다른 10명의 실험대상자를 선정 한 후 첫번째 실험과 동일한 질문을 사용한 똑같은 실험을 반복하였다. 실험결과 첫번째 보다 40%의 에러가 감소하였는데, 이는 79개의 전체 메뉴 중에서 단지 6개의 메뉴만을 재설계하였으나 결과는 매우 높은 효율의 증가를 나타내었다. 즉 메뉴 시스템은 그것을 실제로 사용할 사용자들이 그들의 입장에서 다시 분류하고 이름지을 때 효율이 향상됨을 알 수 있다.

세번째 실험에서는 두번째 실험에서 사용한 메뉴 시스템에 다음과 같은 세가지 에러를 각각 4개씩 삽입한 후 Telidon에 경험이 없는 또 다른 21명의 실험대상자를 선정하여 첫번째 실험을 반복하였다.

-miscategorized item: 특정 메뉴항목을 전혀 관계없는 카테고리에 위치시킴

-synonymous item: 비슷한 메뉴항목을 사용하여 사용자의 선택을 혼동시킴

-vague item: 메뉴항목이 가리키는 내용을 모호하게 함

실험결과 synonymous이나 vague의 경우 어려없이 탐색이 성공할 확률이 각각 45%, 44%였으나 miscategorized된 경우 어려없이 탐색이 성공할 확률은 1%였다. 즉 메뉴 시스템에서 잘못된 메뉴항목의 분류는 정보검색 성공률에 심각한 영향을 미치게 됨을 알 수 있다.

〈표 1〉 exhaustive 탐색시의 최적 메뉴항목 수

		c = 0.50	c = 0.60	c = 0.90	c = 1.35초
k = 0.50초	t = 0.25초	6	6	6/7	7
	t = 0.50	4	4	5	5/6
	t = 1.00	4	4	4	4
	t = 2.00	3	3	3	4
k = 1.00	t = 0.25	7	7	7/8	8
	t = 0.50	5	5	5	6
	t = 1.00	4	4	4	5
	t = 2.00	3	3	4	4

〈표 2〉 self-terminating 탐색시의 최적 메뉴항목 수

		c = 0.50	c = 0.60	c = 0.90	c = 1.35초
k = 0.50초	t = 0.25초	8	8	9	11
	t = 0.50	6	6	6/7	7
	t = 1.00	4	4	5	5/6
	t = 2.00	4	4	4	4
k = 1.00	t = 0.25	10	10	11	12/13
	t = 0.50	7	7	7/8	8
	t = 1.00	5	5	5	6
	t = 2.00	4	4	4	5

Lee와 Whalen은 위와 유사한 몇차례의 실험을 더 한후 메뉴 시스템은 메뉴를 재설계(reclassify & rename)함으로써 현저한 성능향상을 이룰 수 있으며, 재설계는 시스템 설계자가 아닌 실제 사용자가 그들의 시각에서 수행하여야 한다는 결론에 도달하였다.

3.4 메뉴항목의 순서와 선택

사용자들이 메뉴항목이 나열되는 형태에 대한 예측을 할 수 있다면 전체 메뉴항목들을 일일이 읽어볼 필요가 없이 self-terminating 탐색을 할 수 있으므로 메뉴 탐색시간을 줄일 수 있다. 따라서 메뉴항목은 어떤 정해진 순서에 따라 나열하는 것이 바람직하다. 메뉴항목을 나열하는 방법으로는 알파벳 혹은 가나다 순으로 나열하거나 생성순 혹은 그 역순, 사용순서, 사용빈도 혹은 중요도, 포괄적인 개념의 메뉴항목은 어떤 특정한 개념을 지칭하는 메뉴항목보다 먼저 나타나게 나열하는 등 여러가지 방법을 사용할 수 있다. 또한 일반적으로 사용자들은 화면을 좌측상단에서부터 우측 하단으로 탐색해가는 경향이 있으므로, 사용자들이 선택할 가능성이 가장 높은 메뉴항목을 좌측상단에 디스플레이하는 것이 바람직하다.

메뉴에서 원하는 메뉴항목을 선택하는 방법으로는 각 메뉴항목 마다 숫자를 부여하여 원하는 메뉴항목의 숫자를 입력하므로써 메뉴항목을 선택하거나, 영문의 경우 각 메뉴항목마다 문자 하나를 대응시켜 원하는 메뉴항목의 문자를 입력하므로써 메뉴항목을 선택한다. 또한 키보드의 커서 키(→, ←, ↑, ↓)나 마우스를

사용하여 포인터를 원하는 메뉴항목으로 이동시켜 메뉴항목을 선택할 수도 있다. 일반적으로 메뉴항목이 시시각각으로 변하는 동적인 메뉴에서는 숫자를 사용하며, 메뉴가 변하지 않고 항상 일정한 정적인 메뉴의 경우에는 문자를 사용한다. 또한 사용자가 특정 메뉴항목을 선택하면 시스템은 선택한 메뉴항목을 표시하는 피드백 정보를 사용자에게 제공하여야 하며 이때 highlighting 기법을 적용하면 더욱 효과적이다.

3.5 메뉴의 탐색

메뉴 시스템은 단지 몇 개의 메뉴로 구성된 것으로부터 상용 비디오텍스 시스템의 경우에는 약 300, 000개의 메뉴 화면으로 이루어진 것까지 그 범위가 다양하며, 메뉴항목의 수 및 메뉴의 깊이가 커질수록 메뉴의 탐색 중에 현재 어느 위치에 있으며 어디로 탐색해 가야할지를 결정하기 힘든 상태가 발생할 수 있다. 따라서 상용 메뉴 시스템에서는 이와 같은 문제를 완화하기 위한 다양한 장치 및 명령들을 제공해야 한다.

Billingsley[1]는 잘 알려진 동물의 특징이나 습관을 메뉴항목으로 사용한, 38개 메뉴화면으로 이루어진 메뉴 시스템을 구성하여 다음과 같은 세 그룹의 실험대상자들에게 특정 동물을 검색하는 실험을 하였다.

- 그룹 1 : 시스템에 대해 아무 사전 지식이 없는 그룹
- 그룹 2 : 동물들이 알파벳 순서로 나열되어 있는 테이블을 이용하여 시스템을 학습한 그룹, 테이블에는 각 동물을 선택하는 메뉴항목의 순서가 나열되어 있음.

- 그룹 3 : 인쇄된 메뉴 map을 이용하여 시스템에 대하여 학습한 그룹, 메뉴 map에는 각 동물의 분류에 사용한 계층구조 및 특성을 요약

그룹 2와 3의 실험대상자들은 실험 시작전 5분 동안 각각 테이블과 map을 학습하였다. 실험결과 아무 사전지식이 없는 것보다는 테이블이나 메뉴 map을 사용한 그룹의 검색 효율이 좋았으며, 특히 map을 사용한 그룹의 효율은 월등히(약 2배 이상) 향상되었다.

한편 Billingsley는 시스템 전체를 서술하는 메뉴 map을 사용하였으나, 실제로 거대한 규모의 상용 시스템 전부를 서술하는 메뉴 map을 제작하여 사용자들에게 제공하기에는 상당한 부담이 있다. 그러나 탐색의 편의를 위하여 몇몇 상용 시스템에서는 제한된 범위의 메뉴 map을 제작하여 사용자들에게 제공하고 있다[15]. 영국의 Prestel 시스템에서는 수천 개의

메뉴들에 대한 상세한 교차참조 인덱스를 34페이지 분량의 메뉴얼로 만들어 사용자들에게 제공하고 있으며, 미국의 CompuServe 시스템에서는 약 1000여 개의 주제에 대한 인덱스와 메뉴구조의 최상위 3개 레벨의 메뉴(36개 메뉴화면)를 서술하는 다이어그램을 제공하고 있다.

메뉴 map은 인쇄된 형태의 map이며, 이 개념을 적용하여 큰 규모의 상용 메뉴 시스템의 map을 CRT 화면상에 나타내기 위하여는 map의 규모를 축소하여야 한다. map의 규모를 축소하기 위하여는 현재 위치하고 있는 메뉴에서 가장 가까운 일부분만을 보여주거나 자세한 내용은 생략하고 주요 탐색경로만 보여주는 두 가지 접근법이 가능하다. fisheye view [5]는 위의 두 방법을 결합하여 표시할 범위를 축소하고 자세한 내용은 생략하여 화면상에 메뉴 map을 디스플레이하는 방법을 말한다. 즉 계층구조의 메뉴에서 현재 위치하고 있는 메뉴, 그것과 가장 가까이 있는 좌우측의 형제 메뉴들과 주요 탐색경로로써 최상위 메뉴에서 현재 위치하고 있는 메뉴까지의 탐색 경로 상에 있는 모든 상위 메뉴들의 메뉴항목을 한 화면에서 볼 수 있도록 하는 방법이다. 이 방법은 어안은 사람의 눈과 달리 180°를 모두 볼 수 있으나 가운데 부분은 자세하고 주변은 희미하게 보듯이, 현재 위치하고 있는 메뉴는 자세히 나타내고, 형제 메뉴들과 탐색경로상의 상위 메뉴들은 단지 메뉴항목만 화면에 디스플레이하므로써 메뉴탐색에 도움을 줄 수 있다는 개념이다. 이를 구현하기 위하여는 많은 메뉴들을 한 화면에 표시해야 하므로 고해상도의 그래픽 화면이 필요하다.

사용자들에게 현재 위치하고 있는 메뉴에 올 때까지 거처온 탐색경로, 즉 history를 제공하면 사용자는 다음 탐색을 위한 결정을 보다 확고히 내릴 수 있을 뿐 아니라 에러가 발생할 경우에는 적당한 위치의 상위 메뉴로 쉽게 되돌아갈 수 있다. Apperley와 field [1]는 각 메뉴화면 마다 거처온 탐색경로에서 선택한 모든 메뉴항목들을 나열하는 history를 보조 메뉴로 하여 원래의 메뉴와 함께 디스플레이하여, 사용자가 메뉴가 아닌 history상의 항목을 선택할 경우에는 그 항목이 속한 메뉴로 되돌아가서 그 위치에서 탐색을 다시 시작하는 방법을 제안하였다. 이 방식은 데이터베이스에서 하나의 정보가 아니라 몇 개의 정보를 결합하여야만 볼 수 있는 문제를 탐색할 경우 매우 효과적이다.

메뉴상의 메뉴항목 중 어떤 이유로 인하여 현재 선택할 수 없는 상태의 메뉴항목은 highlighting기법을 이용하여 사용자가 시각적으로 인지할 수 있도록 하므로 불필요한 메뉴 선택을 예방할 수 있다. 이 방법은 메뉴를 사용하는 상용 워드프로세서나 PC용 소프트웨어 패키지들이 주로 사용하는 방법이며, 공중 데이터베이스 서비스를 제공하는 시스템의 경우 탐색결과 해당 정보가 없는 메뉴항목들에도 적용할 수 있다. 그러나 Lee와 Whalen[11]이 실험을 통하여 보여준 것처럼 정보가 없는 경우에 사용자는 쉽게 정보검색을 포기하고, 이런 경험이 몇차례 거듭되면 시스템에 대한 불신감이 쌓이게 되므로 정보가 없는 메뉴항목은 처음부터 사용자가 볼 수 없도록 메뉴화면에서 삭제하는 것이 바람직하다.

그외에 계층구조의 메뉴를 탐색할 때 설계된 메뉴 탐색경로를 따라가지 않고 임의로 원하는 메뉴로 이동할 수 있도록 하는 명령어는 메뉴의 규모가 크고 복잡한 경우 탐색효율을 높일 수 있으며, 메뉴를 탐색하여 다양한 정보를 검색하기 보다는 항상 특정 메뉴만을 이용하여 시스템이 제공하는 정보 중 일부 분만을 검색하는 사용자의 경우 자주 사용하는 메뉴를 이용하여 임의의 메뉴구조를 구성할 수 있는 명령어는 불필요한 탐색경로를 생략할 수 있어 효율적이다.

IV. 결 론

인적 측면을 고려하지 않은 사용자 인터페이스는 사용자에게 보다 큰 정신적 부담을 주며 잦은 오류를 유발하여 전체적인 효율을 저하시킨다. 특히 공중 데이터베이스 서비스를 제공하는 메뉴방식의 상용 시스템에서는 다단계 메뉴를 디스플레이하기 위하여 많은 메뉴 화면이 필요하므로 메뉴 선택과 디스플레이로 인한 시간 소모가 크다. 따라서 화면과 메뉴의 설계에는 세심한 주의가 필요하다.

본고에서는 메뉴를 사용하는 사용자 인터페이스를 디스플레이되는 화면의 형태, 제공하는 정보 및 메뉴의 구조, 조작 방법의 편리성 및 효율성의 측면에서 분석할 수 있는 여러 선행연구, 이론 및 방법을 조사하고 이를 토대로 화면과 메뉴를 설계하는데 적용할 수 있는 설계지침을 제시하였다. 이 설계 지침은 기존의 사용자 인터페이스를 재설계하여 사용자의 사용효율을 개선하고 더욱 친숙한 시스템을 구성하고자 할 때에도 적용이 가능하다.

참 고 문 헌

[1] Apperley, M. D. and Field, G. E. "A comparative evaluation of menu-based interactive human-computer dialogue techniques". Proceeding of INTERACT, IFIP, London, 1984.

[2] Billingsley, P. A., "Navigation through hierarchical menu structure: Does it help to have a map?", Proceedings of the Human Factors Society 26th Annual Meeting, Santa Monica, Human Factors Society, 1982.

[3] Bouma, H., "Visual reading processes and the quality of text displays", Ergonomic Aspects of Visual Display Terminal, E. Grandjean and E. Vigliani, Taylor and Francis Ltd, London, 1980.

[4] Duchnicky, R. L. and Kolers, P. A., "Readability of text scrolled on visual display terminals as a function of window size", Human Factors, Vol. 25, No. 6, 1983.

[5] Funas, G. W., "Generalized fisheye views", Proceedings of CHI, New York, ACM, 1983.

[6] Gahtz, W. O., Handbook of screen format design, Wellesley Hills, MA, QED Information Sciences, 1985.

[7] Gould, J. D. and Grischkowsky, N., "Dose visual angle of a line of characters affect reading speed?", Human Factors, Vol. 28, No. 2, 1986.

[8] Helander, M., Handbook of Human-Computer Interaction, North-Holland, 1988.

[9] Kruk, R. S. and Muter, P., "Reading of continuous text on video screens", Human Factors, Vol. 26, No. 3, 1984.

[10] Lee, E. and MacGregor, J., "Minimizing user search time in menu retrieval systems", Human Factors, Vol. 27, No. 2, 1985.

[11] Lee, E., Whalen, T., McEwen, S. and Latremouille, S., "Optimizing the design of menu pages for information retrieval", Ergonomics, Vol. 27, No. 10, 1984.

[12] McDonald, J. E., Stone, J. D. and Liebelt, L. S., "Searching for items in menus: The effects of organization and type of target", Proceedings of the Human Factors Society 27th Annual Meeting, Santa Monica, CA, Human Factors Society, 1983.

[13] Paap, K. R. and Roske-Hostrand, R. J., "The optimal number of menu options per panel", Human Factors, Vol. 28, No. 4, 1986.

[14] Poulton, E. C. and Brown, C. H., "Rate of com-

prehension of an existing teleprinter output and of possible alternatives", Journal of Applied Psychology, Vol. 52, 1968.

[15] Shneiderman, B., Designing the user interface, Addison Wesley, 1987.

[16] Snowberry, K., Parkinson, S. R. and Sisson, N., "Computer display menus", Ergonomics, Vol. 26, No. 7, 1983.

[17] Trollip, S. R. and Sales, G., "Readability of computer-generated fill-justified text", Human Factors, Vol. 28, No. 2, 1986.

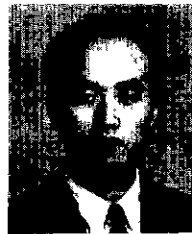
[18] Tullis, T. S., "An evaluation of alphanumeric graphic, and color information displays", Human Factors, Vol. 23, No. 5, 1981.

[19] Tullis, T. S., "the formatting of displays: A review and analysis", Human Factors, Vol. 25, No. 6, 1983.

[20] Tullis, T. S., "Screen Design". Handbook of Human-Computer Interaction, Helander, M. ed., North-Holland, 1988.

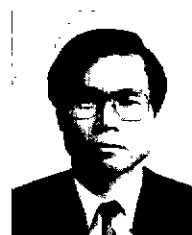
[21] Vartabedian, A. G., "The effects of letter size, case, and generation method on CRT display search time", Human Factors, Vol. 13, No. 3, 1971.

김 혁 만



1985 서울대학교 컴퓨터공학과 졸업
 1987 서울대학교 대학원 컴퓨터공학과 공학석사
 1987 ~ 현재 한국농성 연구개발단 연구원
 1991 ~ 현재 서울대학교 대학원 컴퓨터공학과 박사과정
 관심 분야 데이터베이스 시스템, 철의어 처리, 멀티미디어, 분산 시스템, HCI

이 석 호



1964 연세대학교 정치외교학과 졸업
 1964 ~ 1969 해명장교로 근무
 1975 ~ 1979 미국 텍사스대학교 진신학석사의 박사학위 취득
 1979 ~ 1982 한국과학원 건설학과 조교수
 1982 ~ 1986 한국정보과학회 논문편집위원장
 1982 ~ 1986 한국정보과학회 부회장
 1988 ~ 1989 IBM Watson 연구소 객원교수

1988 ~ 1990 데이터베이스연구회 운영위원장
 1989 ~ 1991 서울대학교 중앙교육연구전산원 원장
 1982 ~ 현재 서울대학교 컴퓨터공학과 교수
 관심 분야 데이터베이스 시스템, 회의처리, 가로구조 등