

# 효율적인 모바일 서비스를 위한 적응적 사용자 인터페이스 기술

## Adaptive User Interface Techniques for Efficient Mobile Service

강영민  
동명대학교

Kang Young-Min  
Tongmyong University

### 요약

모바일 컴퓨팅 분야의 발전에 따라 휴대전화는 더욱 높은 정보처리 기능을 갖춘 스마트 폰(smart phone)으로 변화하고 있지만, 현재의 휴대성 및 사용 용이성을 유지하기 위해 제한된 입출력 기능을 가질 수밖에 없다. 이러한 제한 때문에 사용자 인터페이스 역시 순차적 검색이라는 비효율적인 방식을 사용할 수밖에 없게 한다. 현재의 고정적인 비효율적인 사용자 인터페이스를 극복하고 사용자의 작업 편의성을 높이기 위하여 본 논문에서는 지능적이고 적응적인 사용자 인터페이스를 구현하는 기술을 제안한다. 이 기술은 사용자의 과거 행동 등을 기반으로 지능적으로 변화하는 사용자 인터페이스를 자동 생성함으로써 모바일 장치의 사용 효율을 극대화 할 수 있는 기술이다.

### Abstract

As the mobile computing technologies rapidly develops, the mobile phones are evolving into smart phones capable of higher performance. However, it is inevitable that the mobile phones have limited input/output functions in order to maintain the current portability. Because of the limited input and output, the mobile user interfaces currently adopt the inefficient structures based on sequential search and selection. In this paper, an intelligent and adaptive user interface techniques are proposed in order to surmount the current inefficiency of the static interface and to maximize the user convenience. The proposed techniques, based on the users' previous tasks, automatically generate an intelligent and adaptive user interface in order to make users efficiently utilize the mobile services.

## I. 서론

모바일 장치의 기능이 급격히 고도화되고 제공되는 서비스의 양이 증대함에 따라 모바일 서비스에 대한 편리한 접근을 제공하고 휴대 장치의 활용 효율을 제고하기 위해서는 제한된 입출력 환경에서 최적을 작업 선택과 전환 등을 보장할 수 있는 지능적인 사용자 인터페이스가 요구된다.

지능적인 사용자 인터페이스를 개발하기 위한 노력은 오래된 연구 분야이며, 상황에 따라 적응적으로 변화하는 인터페이스에 관한 연구도 많이 이루어졌다. 적응적인 인터페이스를 개발하는 방법은 다수 사용자의 행동을 바탕으로 얻은 정보를 이용하여 많은 사람이 관심을 가지는 정보를 예측하여 제공하는 방법[1], 특정 개인이 과거에 했던 선택과 거절 이력을 이용하여 콘텐츠를 제공하는 방법[2] 등이 있으며, 메뉴가 스스로 변화하여 사용자에게 적합한 형태로 노출되는 방식[3] 등이 있다.

적응적 인터페이스에 관한 다양한 기법이 연구되었으나 기존의 방법들은 자원이 비교적 풍부한 개인용 컴퓨터 이상의 하드웨어에서 동작하기 위한 것이 대부분이며, 제한된 자원의 모바일 장치에 적응적 인터페이스가 실제로 적용된 예는 아

직 드물다. 본 논문은 휴대전화 장치에서 사용자의 작업을 예측하고 이에 대한 효율적인 접근을 제공하는 지능적이고 적응적인 사용자 인터페이스 기술을 제안한다.

## II. 연구의 목표

기존의 사용자 인터페이스는 메뉴를 정적인 구조로 설계하는 것이 일반적이다. 정적인 구조의 사용자 인터페이스는 특정한 작업을 선택할 때 따라야 하는 길이 고정적이다. 이러한 정적 메뉴 구조의 문제는 설계가 잘못 되었을 경우 매우 비효율적인 작업 경로를 사용자에게 강요할 수 있다는 단점을 가진다. 본 논문에서 제안하는 기법은 빈번한 작업이나 연관이 있는 작업에 빠르게 접근할 수 있도록 사용자의 작업 이력에 따라 메뉴 구조가 적응적으로 변화하는 지능적인 메뉴 인터페이스를 제공한다.

### 1. 기존 인터페이스의 지능적 수준

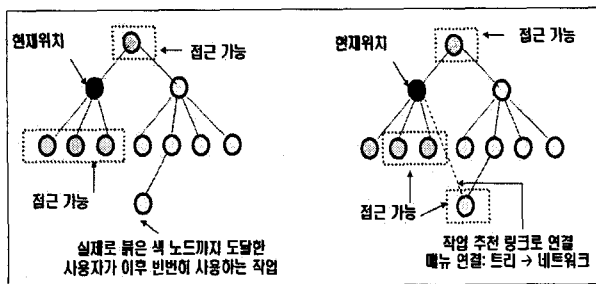
현재 휴대전화 등의 모바일 기기에 적응적인 사용자 인터

페이스가 실제로 적용된 사례는 매우 드물다. 현재 휴대전화 등의 적용되는 사용자 인터페이스는 트리 기반의 정적 구조를 가지며, 이러한 트리 구조를 에지를 따라 탐색하며 선택하는 방식이다. 이러한 고정적 사용자 인터페이스 구조는 설계 방식에 따라 매우 비효율적인 작업 선택을 제공할 수 있으며, 신중한 설계를 통해 구현되었다고 할지라도 사용자들의 다양한 작업 방식에 모두 만족스러운 인터페이스를 제공할 수가 없다. 현재의 휴대전화 사용자 인터페이스가 제공하는 기능적 적용 수준은 직전에 선택한 메뉴 아이템에 디폴트 커서(default cursor)를 두는 정도에 머물러 있다. 휴대전화가 개인의 컴퓨팅 환경에서 차지하는 비중이 지속적으로 증대할 것으로 예상되며, 이에 따라 더욱 복잡해질 모바일 서비스에 사용자가 효율적으로 접근할 수 있게 하는 지능적인 인터페이스가 요구된다.

### 2. 지능적인 인터페이스

휴대전화와 같이 제한적인 입출력을 가진 장치에서 사용자 편의를 높일 수 있는 우수한 사용자 인터페이스는 효과적인 작업 전환, 메뉴의 간결성 유지, 최근 작업 경향에 대한 민감성을 지원해야 한다. 이러한 특성은 정적인 메뉴 구조에서는 얻기 힘든 특성이다.

본 논문은 그림 1에 나타난 것과 같이 현재 위치에서 접근 가능한 노드뿐만 아니라, 실제로 작업을 수행할 가능성이 높은 노드를 추측하여 이를 메뉴에 노출시킬 수 있는 지능적인 사용자 인터페이스를 구현하는 기술을 제안한다. 이러한 적응적 인터페이스는 메뉴의 계층 구조를 정적으로 유지하지 않고 동적으로 변화하는 방식이어야 한다.



▶▶ 그림 1. 연관 작업의 연결을 통한 적응적 UI의 구현

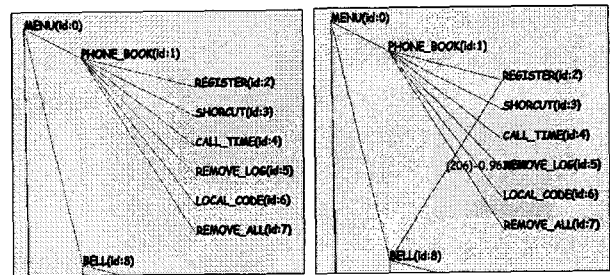
### III. 지능적 사용자 인터페이스 구현 기술

휴대전화의 메뉴를 지능적으로 개선하기 위하여 본 논문은 효율적으로 작업을 전환하고, 최근 작업에 민감하며 메뉴의 간결성을 유지할 수 있는 방법을 제안한다.

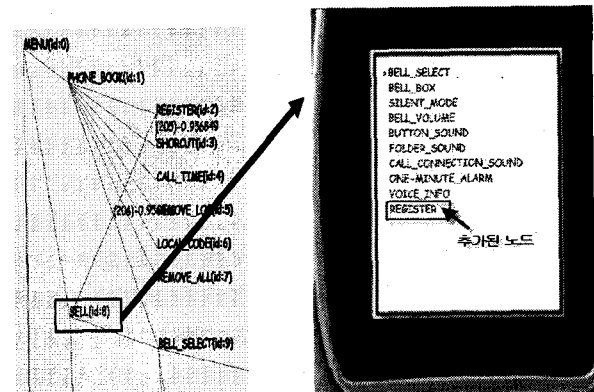
### 1. 효율적인 작업 전환

효율적인 작업 전환을 위해서는 빈번히 선택되는 작업을 파악해야 하며, 작업들 사이의 연관성을 분석해야 한다. 메뉴 트리에 존재하는 다수의 작업 중에서 빈번히 사용되는 작업은 소수이다. 이러한 빈번한 작업들 사이에 연결이 있으면 효율적인 작업 전환이 가능할 것이다. 전체 메뉴에서 빈번히 사용되는 작업 정보뿐 아니라, 작업들 사이의 연관관계도 효율적인 작업 전환의 중요한 정보가 될 수 있다. 예를 들어 알람을 설정하는 작업을 수행한 경우, 이 이후에 일어날 작업은 알람 설정과 밀접한 연관을 갖는 진동모드 해제 작업이나 벨소리 설정 작업일 가능성이 높다. 이러한 작업들 사이의 연관성을 파악함으로써 더욱 효율적인 작업 전환을 제공할 수 있게 된다.

본 논문의 기법은 특정 작업이 수행된 이후 이어서 다른 어떤 작업이 선택될 경우에 이 두 작업이 서로 연관성이 있다고 판단하는 것이다. 그림 2의 (a)와 같은 구조에서 "Register" 메뉴를 선택한 이후 다시 최상위 노드로 이동한 뒤 "Bell"을 선택하였다면, 이 두 동작은 연관성을 가졌다고 판단할 수 있다. 따라서 그림 (b)와 같이 연관성 에지(edge)를 추가하여 바로 작업을 전환할 수 있는 길을 제공하는 것이다. 이러한 연관정보를 이용하면 그림 3과 같이 초기 메뉴와 달리 사용자가 작업을 전환할 가능성이 높은 연관 노드를 메뉴에 노출할 수가 있다.



▶▶ 그림 2. 연관정보가 추가된 메뉴



▶▶ 그림 3. 연관정보에 따라 변화된 메뉴

연속적으로 일어난 두 작업은 실제로 연관성을 가지고 있는 것이 아니라 우연히 연속적으로 나타난 것일 수도 있다. 본 논문의 기법은 이러한 연속적 발생 기록을 지속적으로 축적하여 유의미한 연관 노드의 가중치가 우연히 연속으로 발생한 두 작업의 가중치를 압도하게 될 것이라는 가정을 가지고 구현되었다.

메뉴의 작업 아이템이  $n$ 개인 경우 가능한 모든 연관정보의 수는  $O(n^2)$ 이 된다. 이는 지나치게 많은 정보로 휴대장치에 부담을 주게 된다. 실제로는 유의미한  $n$ 개의 연관정보  $Rel[1...n]$ 만 유지한다. 연관 정보  $Rel[i]$ 는 연관되는 두 노드의 식별자  $Rel[i].node_1$ 과  $Rel[i].node_2$ , 그리고 두 노드의 연관 가중치  $Rel[i].w$ 로 구성되며 전체 배열의 순서는 가중치에 따라 정렬되어 있다. 즉,  $Rel[i]$ 의 가중치를  $Rel[i].w$ 라고 할 때 다음과 같이 표현할 수 있다.

$$i < j \Leftrightarrow Rel[i].w < Rel[j].w \quad (1)$$

어떤 두 작업 노드  $n_i$ 와  $n_j$ 가 연속적으로 선택되었다면 우선 이 두 노드의 연관정보가 배열에 저장되어 있는지를 검사한다. 만약 이 정보가  $n$ 개의 연관정보 배열 가운데 하나인  $Rel[k]$ 로 존재한다면 그 연관정보의 가중치를  $\alpha$ 만큼 증가시킨 뒤,  $Rel[k]$ 가 전체 배열에서 적절한 위치에 자리 잡도록 이동을 시킨 후, 전체 연관 정보 각각의 가중치를  $\alpha/n$ 만큼 감소시킴으로써 전체 가중치 합을 일정한 상수로 유지한다. 만약 노드  $n_i$ 와  $n_j$ 의 연관성이 연관정보 배열에 존재하지 않는다면 첫 번째 연관정보를  $Rel[1].node_1 = n_i$ ,  $Rel[1].node_2 = n_j$ 로 설정하고 이 앞서 사용한 방법과 같이 가중치를 증가시킨 뒤 전체 가중치가 원래의 상수가 되도록 조정한다. 사용자의 오랜 작업 이력을 활용하여 연관 정보를 갱신해 나가면 연관정보를 얻을 수 있으며, 연관성이 높은 연관노드를 노출함으로써 지능적이고 적응적인 메뉴 구성이 가능하다.

## 2. 하위 아이템의 우선순위

메뉴에 나타나는 아이템은 기본적으로 메뉴 트리 상에 존재하는 현재 노드의 자식 노드들이다. 일반적인 메뉴 인터페이스는 최초로 설계된 고정적인 메뉴 트리를 그대로 보여준다. 이 방식은 실제 사용 빈도를 반영하지 못한다. 본 논문은 선택 가능성이 높은 자식 노드를 우선적으로 노출하여 상위에 두는 방식을 사용한다.

자식 노드의 우선순위를 설정하는 방법은 방문 횟수를 기반으로 많은 방문이 이루어졌던 노드를 우선적으로 보이게

하는 방법과 최근에 사용되었던 노드를 우선적으로 보이는 방법으로 구분할 수 있다. 방문 횟수를 기반으로 한 방법은 특정한 노드를 매우 오랜 기간 여러 번 방문한 뒤에는 사용자가 새로운 작업 패턴을 보일지라도 이를 반영하기가 쉽지 않고, 다시 오랜 시간이 필요하다는 단점을 가진다. 즉, 최근 작업에 대한 민감성이 크게 떨어지는 것이다.

최근 방문 노드에 민감한 우선순위 설정 방법은 방문순서 역순으로 보여주는 것이다. 어떤 메뉴의 하위 노드가  $a_1, a_2, a_3, a_4$ 이고, 지금까지 노드를 방문한 이력이  $a_1, a_2, a_3, a_2, a_2, a_2, a_3, a_3, a_4, a_1$ 인 경우 메뉴 노출 순서를  $a_1 \rightarrow a_4 \rightarrow a_3 \rightarrow a_2$ 로 보여 주는 것이다. 이 방법은 최근 방문 노드에 민감하다는 장점을 가지지만, 노출 순서만 존재하므로 노드 상호간의 중요도 비교가 불가능하다.

본 논문은 최근 방문 노드에 민감하면서 수치화된 가중치를 제안함으로써 최근 방문 노드 역순으로 메뉴를 노출하는 방식의 장점을 그대로 유지하면서도 노드 각각의 중요도를 정량적으로 판단할 수 있는 기법을 제안한다.

우선 메뉴가 처음으로 사용되는 초기에 자식 노드들은 모두 동일한 가중치로 1의 값을 가진다. 자식 노드들은 이 가중치에 따라 정렬되어 노출되므로, 어떤 노드가 선택되면 그 노드의 가중치가 증가하고 다른 노드들은 감소한다. 새롭게 설정된 가중치에 따라 노드를 다시 정렬함으로써 노드의 노출 우선순위가 변화된다. 이때 가중치를 조정하는 방식을 어떻게 설계하느냐에 따라 이전 작업에 대한 민감도가 결정된다. 본 논문은 이전 작업에 극도로 민감한 모델을 제안한다. 이 모델을 필요에 따라 민감도를 조절할 수 있다.

자식 노드의 개수가  $n$ 개일 때, 각각의 노드  $i$ 는 가중치  $w_i$ 를 가진다. 자식 노드 가중치의 총합은 상수  $n$ 이 된다. 즉,

$$\sum_{i=1}^n w_i = n \quad (2)$$

어떤 노드  $k$ 가 선택되면 식 (2)는 그대로 유지하면서  $w_k$ 는 증가하고 다른 모든 노드의 가중치를 감소하는 변화가 일어나야 한다. 가중치 변화를 수행하기 위해 가장 먼저 수행하는 작업은 선택된 노드  $k$ 가 아닌 모든 노드  $i$ 에 대해 가중치를 다음과 같이 일정한 비율을 감소시키는 것이다.

$$w'_i = w_i - w_i/\alpha \quad (\alpha > 0) \quad (3)$$

다음으로 수행해야 하는 작업은  $w_k$ 의 값을 증가시키는 것

이다.  $w_k$ 의 증가가 끝난 뒤에 전체 가중치의 합은 식 (2)와 같이 여전히 상수로 남아있어야 하므로 다음과 같이 증가를 수행한다.

$$w_k = w_k + \sum_{i \neq k} \frac{w_i}{\alpha} \quad (4)$$

이러한 작업이 실제로 수행되기 위해서는  $\alpha$ 의 값을 결정해야 한다. 이 값에 따라 우선순위 결정 방법의 특성이 좌우된다. 본 논문은 최근 작업에 극도로 민감하여 최근에 선택된 노드가 반드시 가장 높은 우선순위를 갖도록 하는  $\alpha$  값을 사용하였다. 이러한  $\alpha$  값을 찾는 방법을 설명할 것이다.

가중치는 내림차순으로 정렬된다. 따라서 가장 큰 가중치는  $w_1$ 이며, 가장 작은 가중치는  $w_n$ 이 된다. 우리가 구하는  $\alpha$  값은 가장 작은 가중치  $w_n$ 을 갖는 노드가 선택되었을 경우에도 바로 1위의 가중치로 변화할 수 있도록 하기에 충분한 값을 사용해야 한다.  $w_n$ 이 증가하여 얻게 되는 새로운 가중치  $w'_n$ 은 다음과 같다.

$$w'_n = w_n + \sum_{i=1}^{n-1} \frac{w_i}{\alpha} \quad (5)$$

이 값은  $w_1$ 이 감소하여 얻게 되는  $w'_1$ 보다 커야만 한다.  $w'_1$ 은 식 (3)과 같다. 따라서 우리는 다음과 같은 부등식을 얻는다.

$$w_n + \sum_{i=1}^{n-1} \frac{w_i}{\alpha} > w_1 - \frac{w_1}{\alpha} \quad (6)$$

식 (6)을  $\alpha$ 에 대하여 정리하면 다음과 같은 결과를 얻는다.

$$\begin{aligned} \alpha w_n + \sum_{i=1}^{n-1} w_i &> \alpha w_1 - w_1 \\ \alpha (w_n - w_1) &> -(\sum_{i=1}^{n-1} w_i + w_1) \\ \alpha (w_1 - w_n) &< \sum_{i=1}^{n-1} w_i + w_1 \\ \alpha &< \frac{\sum_{i=1}^{n-1} w_i + w_1}{w_1 - w_n} \\ \alpha &< \frac{\sum_{i=1}^n w_i + w_1 - w_n}{w_1 - w_n} = 1 + \frac{\sum_{i=1}^n w_i}{w_1 - w_n} \end{aligned} \quad (7)$$

식 (7)과 식 (2)에 의해  $\alpha$ 는 다음과 같은 상한을 갖는다

$$\alpha < 1 + \frac{n}{w_1 - w_n} \quad (8)$$

식 (8)과 같이  $\alpha$ 를 설정하면 선택된 노드의 가중치가 1위를 차지하게 된다. 그러나 이 식은 모든 노드가 동일한 가중치를 갖는 상황에서  $w_1 - w_n$ 이 0이 되어 문제가 발생한다. 또한 이 값이 0이 아니더라도 가중치 차이가 크지 않은 경우 지나치게 큰 값이 된다. 식 (8)은  $\alpha$ 의 상한을 설정하는 것이므로, 본 논문에서는 이 값보다 더 작고 안전한 다음과 같은 상한을 사용하였다.

$$\alpha < 1 + \frac{n}{1 + w_1 - w_n} \quad (9)$$

$\alpha$ 를 통해 가중치를 감소시켰을 때에, 그 값이 여전히 양수이어야 하기 때문에  $w_i - w_i/\alpha > 0$ 라는 조건을 만족하여야 하며, 이를 통해  $\alpha$ 의 하한이 1이라는 것, 즉  $\alpha > 1$ 을 쉽게 알 수 있다. 식 (9)의 부등호 우변에 있는 값은 명백히 1보다 크므로 우리는 다음과 같이  $\alpha$ 를 결정할 수 있다.

$$\alpha = 1 + \frac{n}{1 + w_1 - w_n} \quad (10)$$

본 논문에서 사용된 가중치 모델을 적용했을 때 얻게 되는 자식 노드의 순서는 최근 방문 노드의 순서와 일치한다. 그러나 이 기법이 최근 방문 순서만을 고려하는 방법에 비해 나은 점은 단순한 선후 관계뿐 아니라 노드의 중요도를 정량적으로 상호 비교할 수 있으며, 필요한 경우 자주 쓰지 않는 노드를 판별하여 간략화된 메뉴를 제공할 수 있다는 것이다. 또한 수치화된 노드 우선순위는 연관성 정보와 비교가 가능하기 때문에 자식 노드와 그림 3에서 볼 수 있는 것과 같은 연관 노드를 동시에 보여줄 때, 상호 비교하여 연관 노드가 자주 쓰지 않는 자식 노드보다 먼저 노출되도록 할 수도 있다는 것이다.

#### IV. 실험 결과

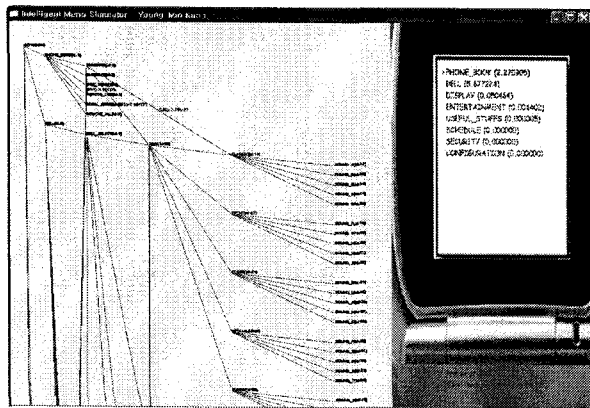
그림 4는 본 논문의 실험을 위해 구현된 시스템이다. 화면의 왼쪽 부분은 메뉴 계층 구조가 나타나며, 사용자의 작업 이력에 따라 추가된 연관정보 등을 보이고 있다. 오른쪽 화면에는 현재 상태에서 사용자에게 실제로 보이는 메뉴가 나타

나 있다. 그림 5는 연관정보와 노드 가중치 정보를 이용하여 실제로 적응적인 메뉴 배치를 실행한 결과이다. 그림 5에서 사용자는 'Entertainment' 메뉴 노드에 있으며, 이때 하위 메뉴는 'Game', 'Karaoke', 그리고 'Song\_Download'이다. 오른쪽 화면을 보면, 사용자는 세 개의 하위 메뉴뿐 아니라, 연관된 3 개의 작업에 바로 갈 수 있는 메뉴를 볼 수 있다. 이 메뉴의 노출 순서는 가중치에 따라 새로 정렬될 수도 있다.

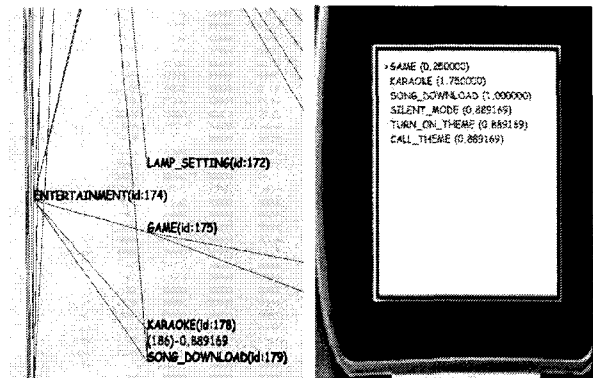
V. 결론

본 논문은 휴대전화 장치에 적용할 수 있는 지능적이고 적응적인 사용자 인터페이스를 구현하는 기술을 제안하였다. 제안된 기법은 제한적인 자원을 가지고 있는 휴대전화에 적용할 수 있도록 사용자의 전화기 사용 이력을 로그(log)로 저장하거나 대규모의 메모리를 요구하는 방식을 피하고  $O(n)$  개의 노드를 가진 메뉴 트리에  $O(n)$  개의 추가적 에지(edge)와  $O(n)$  개의 가중치 정보를 이용하여 적응적으로 변화하는 사용자 인터페이스를 생성할 수 있다.

본 논문의 기법을 바탕으로 휴대전화 장치에 지능적으로 변화하는 사용자 인터페이스 기술이 폭넓게 적용될 경우 사용자의 휴대전화의 사용 효율을 극대화할 수 있을 것이며, 휴대전화를 이용하여 다양하고 복잡한 작업을 수행할 때 발생하는 사용자 부담을 최소화하여 더욱 높은 수준의 서비스 요구를 창출할 것이다.



▶▶ 그림 4. 실험을 위해 구현된 시스템



▶▶ 그림 5. 적응적 UI 결과: 연관 정보와 노드 가중치에 따라 변화한 사용자 인터페이스

감사의 글

본 논문은 2006년도 SK Telecom 재원으로 설립된 동명대학교 SKTU차세대통신기술연구소 학술연구비 지원에 의하여 이루어진 것입니다. (SKTU-06-002)

참고 문헌

- [1] Langley, P. User modeling in adaptive interfaces. Proceedings of the Seventh International Conference on User Modeling, 357-370.
- [2] Ross, E. Intelligent User Interface: Survey and Research Directions. Technical Report CSTR-00-004, Department of Computer Science, University of Bristol.
- [3] Sears, A. and Shneiderman, B., "Split menus: Effectively using selection frequency to organize menus," ACM Transactions on Computer-Human Interaction, 27-51.