

스키마 개념을 도입한 사용자 계획수립의 용이도 평가

윤완철*

Evaluation of Planning Transparency of User Interface Reflecting State Schemas

Wan Chul Yoon*

Abstract

It becomes increasingly important to design user interface to carry low complexity. The cognitive limitations of users severely restrict utility of highly intelligent but complex modern systems. Since humans are known to use schemas to reduce cognitive complexity, imposing good consistency to an interface design that may help the user form useful schemas will provide powerful control over the complexity.

This report presents a research effort to develop a quantitative method for evaluating interface complexity that the user would experience planning his or her course of action. Taking into account the user's potential schemas, a quantitative measure based on information theory was developed to assess the navigational complexity. This approach does not rely on the subjective judgment of the researcher as most schemes dealing with user schemas do. The proposed method may benefit the rapid prototyping approach to design a better user interface by allowing handy assessment of the design.

1. 서론

대규모화와 고도의 자동화, 지능화의 속성을 지닌 현대적 시스템들에서는 점차로 인간의 운

용능력이 중요한 문제점으로 부각되고 있다. 이것은 그 대규모화와 자동화추세에 필연적으로 따르는 시스템의 복잡도 증가가 인간의 문제해결 및 의사결정 능력의 한계를 위협하는데에 기인한다. 한 때 자동화 또는 지능화가

* 한국과학기술원 산업공학과

인간의 약점을 극복하는 자연스러운 해결방향이라고 믿는 경향도 존재했으나, 이제는 인간을 적극적으로 고려하여 시스템과 그 안의 자동 또는 지능적 기능을 설계하지 않는 한, 자동화나 지능화 그 자체가 인간의 기능을 높이고 전체 시스템의 운용성을 증진하는 것은 아니라는 인식이 경험을 통해 확산되고 있다.

지능적인 인터페이스란 인간의 편의를 고려할 뿐 아니라 인간의 기능을 극대화시킬 수 있어야 한다. 이것은 사람이 주관적으로 경험하는 시스템사용의 복잡성, 즉 인터페이스의 인지적 복잡도(Cognitive Complexity)를 제어하는 문제로 귀착된다. 시스템 자체의 복잡도와 사용자 인터페이스의 복잡도는 단순히 비례하지 않으며, 따라서 사용자의 입장에서 인터페이스 복잡도를 분석하고 평가하는 작업은 시스템의 성과와 효용을 보장하는데 가장 중요한 요소의 하나로 인식되고 있다.

시스템의 구축 후에 실험적으로 사용자 복잡도를 측정하는 일은 비교적 용이하나 그 결과로부터 시스템을 개선하기는 매우 어렵고 근본적인 개선이 불가능할 수도 있다. 따라서 인터페이스의 설계는 시스템의 제작에 착수하기 전에 충분히 검토되어야 하며, 그것은 곧 평가와 개선을 반복하는 과정에 의존하므로 개념적 설계 단계에서 사용자복잡도를 평가하는 기법의 개발은 매우 큰 중요성을 지닌다 할 수 있다.

본 논문은 사람이 소프트웨어나 전자기기, 기타의 인터페이스를 접할 때 필연적으로 수행하는 계획수립을 그 인터페이스의 구조가 얼마나 용이하게 해 주는 성질이 있는가를 평가하기 위한 정량적 방법을 모색하는 것이다. 먼저 이 문제의 범위와 그 중요성을 전체 인터페이스 운용의 구조에 비추어 논하고, 이어 이 문제의 맥락에서 적용될 필요가 있는 스키마의

개념을 정리한 뒤, 엔트로피의 개념에 입각한 복잡도 평가의 방법론을 제시한다.

2. 시스템 상태에 관한 지식과 인터페이스 계획수립

인간이 인터페이스의 사용에서 겪게 되는 주관적 복잡도를 어떻게 예측하느냐 하는 문제에는 일반적인 해답이 주어지고 있지 않다. 그 가장 중요한 원인은 사용자의 경험하는 복잡도가 시스템의 구조와 기능의 구성만으로 결정되는 것이 아니라, 사용자가 지닌 지식의 구조에 의해 달라지기 때문이다[4, 5].

일반적으로 어떤 문제의 복잡도의 평가에 사용될 수 있는 방법은 여러 가지가 있다[3]. 첫째, 정보이론적인 측정치를 사용하는 방식이 있는데, Rouse 등은 피훈련생들의 진단문제에서의 성적을 대상 시스템의 구조에 대한 정보측정치를 근거로 하여 매우 정확하게 예측하는데 성공하였다[5]. 이 종류의 측정치는 인터페이스의 구조적 복잡도를 평가하는 용도에 주로 유용하리라 예상할 수 있다.

복잡도 측정의 다른 방식은 주어진 문제를 해결하는 알고리즘의 계산상의 복잡도를 측정치로 사용하는 것으로서 Combinatorics 등에서 흔히 사용된다. 그러나 사용자 인터페이스의 경우에 응용하기 위하여는 그 알고리즘이 곧 사람이 문제해결 과정을 반영하는 것이라야 한다는 전제가 따른다. Card, Moran과 Newell [2]이 제안한 GOMS모형은 문제해결 과정을 규칙들로 표현하고 그 규칙의 수를 복잡도 측정에 이용하므로 바로 이 범주에 드는 복잡도 측정방식이라 할 수 있으나, 복잡한 문제에서

의 인간의 사고과정에 대한 묘사의 근접도와 객관성에서 부족함이 있다.

세째로 조작 대상 시스템의 구성상의 복잡도를 그대로 사용하는 비교적 단순한 방법도 많은 경우 유용하나, 이는 인터페이스 복잡도의 연구가 시스템 내적구조와 보유기능의 복잡성에 비해 인터페이스의 복잡성을 감소시키고자 하는 것이라는 목적에 상치되는 전제를 가정하게 되므로 적용하기 곤란하다.

본 논문에서는 인터페이스의 사용에 있어 사람이 수행해야 하는 계획의 수립에 토대가 되는 인터페이스의 구조적 지식에 중점을 두고 그 복잡도를 평가하려 한다. 구조적인 지식은 시스템 상태와 조작의 종류, 그리고 그 양자간의 상호관계에 관한 지식으로 이루어 지는 것으로 정의한다. 이는 General Problem Solver등에서 계획수립작업(Planning)에서의 탐색공간(Search Space)이 상태(State)와 조작(Operation)들로 이루어 지는 것과 같이, 인터페이스의 보다 기능적인 지식을 습득하고 활용하는 데 구조적 배경을 형성하는 지식이 된다고 할 수 있다. 사람이 어떤 조작을 수행하면 그에 따라 대상 시스템은 정해져 있는 모종의 행동을 한 후 스스로의 상태(System State)를 변환시킨다. 사람에게 의한 후속 조작의 가능성은 이 결과적 상태에 의해 제약받게 된다. 따라서 사람에게 의한 조작 계획의 수립작업은 이렇듯 상태의 변화와 각 중간 상태에서 가능한 조작의 연쇄를 구성하는 것을 그 구조로 한다. 이 구조적 면에 대비되는 것은 기능적 면으로서 각 조작이 야기하는 시스템의 행태의 내용이라 할 수 있다.

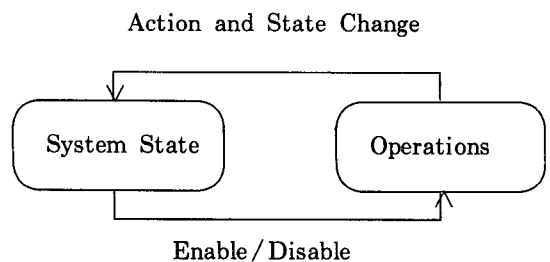
즉, 본 논문에서 다루는 인터페이스 계획수립의 구조적 면이란 Navigation의 개념에 해당한다. 이 Navigation의 개념을 도식화하면 [그림 1]과 같다. 이 그림에서 사람에게 의한 조

작이 유발하는 시스템의 행위나 상태의 변화는 그 시스템에 대한 기능적인 이해의 범주에 들어가지만, 시스템 상태에 의하여 제약되거나 가능하여지는 후속조작의 집합은 기능과는 별도로 거의 순전히 계획수립을 위하여 사용되는 지식이라 할 수 있다. 이 Navigation구조에 대한 지식의 복잡도는 사용자가 인터페이스 계획을 수립할 때 경험하게 되는 인지적 복잡도에 매우 직접적인 영향을 미치게 된다.

3. 사용자에게 의한 스키마의 사용

사용자의 주관적 복잡도를 평가하기 위하여는 대상 시스템이나 인터페이스 자체에 관한 표현보다는 그 인터페이스를 사용하기 위하여 요구되는 사용자의 지식에 대한 표현에 엄밀성을 기하여야 한다. 만일 표현된 지식이 인터페이스 자체를 정확히 묘사했으나 사람의 지식습득 및 활용방법과 다른 특성을 가지고 있다면, 그러한 분석을 통하여 사람이 경험하는 복잡도는 예측될 수 없을 것이다.

사람에게 있어 지식습득과 활용에서의 인지적 복잡도를 줄이고 실수나 학습오류를 줄여 줄 수 있는 가장 강력한 수단은 대상 지식에



[그림 1] 인터페이스 계획의 Navigation 구조

내재된 일관성(Consistency)을 파악하는 것이다. 이러한 일관성은 원형(Prototype)의 발생을 촉진시킨다. 또 이 원형으로부터 각 개별 지식단위의 편차가 적을수록 일관성은 커진다고 할 수 있다. 이러한 일관성 또는 유사성의 파악과 활용은 인간의 인지활동에 있어 스키마(Schemas)의 구조를 통하여 일어난다는 것이 알려져 있다[Ande80.]

즉, 스키마는 소속 개체의 여러 특성으로부터 귀납되는 개념적인 원형이며 각 개체는 해당 스키마에 대하여 원근의 관계를 가진다. 따라서 스키마의 파악은 각 개체를 모두 독립적으로 이해하고 기억해야 하는 비경제성을 면하게 해 준다. 이는 다시 인간으로 하여금 복잡한 대단위 시스템을 쉽게 이해하고 사용하는 데 도움을 준다. 즉, 스키마의 사용은 새로운 지식의 이해와 사용에도 기존지식의 구조를 사용하게 함으로써 경제성을 부여한다. 그러므로 인터페이스에 대한 인간의 주관적 복잡도를 측정하기 위한 분석과정에 반드시 고려되어야 하는 것이 스키마 구조라 할 수 있다.

인터페이스 지식에서 스키마구조는 다양한 단면에서 발생함이 관찰된다. 그러나 위에서 논한대로 조작계획의 구조적 측면에 직접 관련되는 스키마는 어떤 상황에서 가능한 작업과 그 경로의 집합에서 나타나는 유사성이다. 이 스키마의 개체는 시스템 상태들이 된다.

4. 인터페이스의 구조적 복잡도 평가

4.1 시스템상태의 조작적 정의

인터페이스 지식의 내용은 목적에 따른 구동 절차, 시스템의 동작원리 또는 모형, 사람에 의해 시동되는 조작기능들의 종류와 그 구문론, 각 조작에 반응하는 시스템의 행동, 실현 가능한 시스템의 상태들과 그 상태들 사이의 이동 방법, 상태와 조작 사이의 관계등의 여러 양식으로 나타내어 진다. 전술한대로 본 연구에서는 인터페이스 지식을 구조적인 지식과 기능적인 지식으로 구분하여 그 중 구조적인 지식의 측면을 다룬다. 이는 계획수립과정에 배경이 되는 Navigation지식, 즉 각 시스템 상태에서 가능하게 되는 조작들이 무엇인가 하는 지식을 사용자가 얼마나 쉽게 습득하고 활용할 수 있는가를 평가하는 문제로 귀결된다.

그러나 이 분석을 위하여 시스템 상태를 정의해 나가는 방식은 General Problem Solver나 시뮬레이션 등 시스템모형들에서의 방식과 상이할 필요가 있다. GPS등에서는 조작들이 시스템상태 사이의 이동과 대응되는 것으로 표현되고, 시스템상태들은 어떤 작업을 수행하는 각 단계를 구성하는 것으로 파악된다. 즉 시스템상태가 시스템의 내적 동작원리들에 의하여 먼저 정의되고 조작들은 그들 사이의 이동에 대응되어 정의되는 것이다. 그러나 현대적 시스템들에서는 시스템의 내적 동작원리를 사용자가 이해하고 의식한다는 가정을 할 수 없다. 그러므로 사용자 지식의 표현과 분석에 이러한 동작원리에 입각한 시스템 상태의 정의를 사용하는 것 역시 합리적이라고 할 수 없다.

그러므로 본 연구와 같은 사용자 지식을 대상으로 하는 연구에서는 사용자에게 허용된 조작들을 기반으로 하여 그 조작들을 이해하기 위하여 상정해야 하는 시스템 상태들을 정의하고, 이들로 이루어지는 최소집합을 점차 확장해 나아가야 할 것으로 사료된다. 사용자가 어

<표 1> VCR의 상태-조작-관련표

State Opr	Go Stop	Go Play	Go Record	Go FF	Go Rew	Go F.Scan	Go B.Scan
Stop	N	P	P	P	P	N	N
Play	P	N	N	N	N	P	P
Record	P	N	N	N	N	N	N
FF	P	P	P	N	P	N	P
REW	P	P	P	P	N	P	N

면 상황에서 다른 때 불가능하였던 새로운 조작이 가능함을 발견하였을 때, 그는 그 상황을 그 이전의 상황과 구분하고 어떻게 다른가를 파악하려 한다. 반대로 어떤 조작이 불가능하게 되었을 때에도 같은 일을 한다. 예를 들어 Macintosh 컴퓨터의 소프트웨어들에서는 Clipboard에 복사된 내용이 있는 상태에서만 Paste조작이 가능해지므로 Clipboard의 상태는 시스템상태를 형성하는 요소가 된다. 시스템을 배우는 사용자의 입장에서는 사용가능한 조작을 중심으로 시스템상태를 인식하게 되는 것이다. 이와 대조적으로 시스템의 내적인 상태는 반드시 사용자에게 인식될 필요가 있는 것이 아니므로 사용자 지식의 모형에 포함되지 않거나 전문가급 사용자를 위한 보조적 모형에 포함되는 것이 정당할 것이다. 예를 들어 일반 가정용 VTR의 상태들과 그에 따라 가능한 조작들은 다음 <표 1>과 같이 나타낼 수 있다.

이와 같이 시스템 상태가 관련되는 조작의 집합에 대응되도록 정의되면, 관련되는 모든 조작에 있어 동일한 두 시스템 상태는 사용자의 입장에서는 구별되지 않고 하나의 그룹으로 인지될 수 있다는 점이 반영되는 것이다. 위 표의 경우, 우리가 실제 느끼는 Record와 Play 상태사이, 또 FF와 REW상태 사이의 유사성이 해당 열끼리의 유사성으로 반영되어 있음을 볼

수 있을 것이다. 이 방식의 도입으로써 내부동작이 복잡한 반면 인터페이스는 상대적으로 단순한 현대적 시스템들의 인터페이스 묘사가 불필요한 복잡성을 피할 수 있으리라 생각된다. 또한 시스템 상태의 이러한 정의는 객관적이고 간략하다는 장점이 있을 뿐 아니라 사용자가 인터페이스를 접할 때 가장 상위 개념에서 인식하게 되는 상황적 구분과 잘 일치하는 것으로 관찰된다.

4.2 상태-조작 관련표의 엔트로피 (Entropy)

이 상태-조작 관련표가 지니는 사용자 지식의 복잡도를 정량화하기 위해서는 앞에서 논의된 바와 같이 정보이론적인 측정치가 가장 유효할 것이다. 이는 곧 시스템의 어떤 상태에서 사용자가 한 조작의 가능 여부를 알기 위해 필요로 하는 정보의 양을 가리키며 본 논문에서는 그 정보량을 엔트로피개념을 도입하여 산출하는 방법을 취한다.

이제 어떤 시스템에 조작(Operation)의 집합 $O = \{O_1, O_2, \dots, O_r\}$ 이 있고 O의 서로 다른 부분 집합을 가능케 하는 State의 집합 $S = \{S_1, S_2, \dots, S_n\}$ 이 있다고 하자. O와 S사이에는 다음과

〈표 2〉 State-Operation Contingency Table

	O ₁	O ₂	O ₃	O ₄	O ₅
S ₁	0	0	1	1	1
S ₂	1	1	1	0	0
S ₃	1	1	0	0	0
S ₄	1	0	1	0	0

같은 Contingency Matrix가 존재한다.

$$A = \{a_{ij} \mid i = 1, \dots, n; j = 1, \dots, r\},$$

$$a_{ij} = 1, \text{ if } o_j \text{ is available in } s_i$$

$$a_{ij} = 0, \text{ otherwise}$$

일례로 〈표 2〉와 같은 행렬 A가 나타날 수 있다.

이제 사용자가 한 조작의 사용 가능성을 어떤 상황에서 알기 위한 노력, 또는 한 조작에 연관시켜 시스템의 상황들을 파악하기 위한 노력의 양을 엔트로피개념을 도입하여 정량화해보면 다음과 같이 된다. 한 시스템상태에서 그 조작이 가능한지의 여부에 관련된 정보의 양은

$$I(a_{ij}) = \log\{1/P(a_{ij})\}$$

여기서 P(a_{ij})는 어느 쪽이든 a_{ij}의 값이 실현될 확률이다. 즉 실제로 나타난 값이 작은 확률을 가진 것일수록 그 실현에 의한 정보의 양은 커지게 된다. 조작 j에 관련된 정보의 총량은 각 상태에서의 정보량의 합계로서

$$I(O_j) = \sum_i I(a_{ij}) = \sum_i \log\{1/P(a_{ij})\}$$

이다. 이 정보량은 시스템상태의 수가 증가함에 따라 단조 증가한다. 그러나 시스템상태 i에서 a_{ij}의 값이 정확히 예측될수록 P(a_{ij})는 1에

가까워지고 그 시스템상태로 인한 정보량의 증분은 0에 가까워지게 된다. 즉, 사용자가 시스템상태 i에서의 조작 j의 가능성 여부를 주변 정황으로부터 강하게 예측할 수 있는 경우에는 그로 인한 복잡도의 증가는 그 만큼 적어질 수 있음을 나타내는 것이다. 나아가서 어느 조작이 각 시스템상태에서 어떤 가능성을 갖는지 잘 예측할 수 있게 하는 구조가 존재하는 경우에는 그 조작으로 인한 전체 불확정성이 작아지고 학습노력도 감소할 것으로 예측할 수 있다. 이러한 구조는 곧 전장에서 논의한 지식의 스키마적 구조의 발현이며 계속되는 논의에서와 같이 유사지수(Similarity Index)의 정의를 통해 정량적으로 다루어 질 수 있다.

a_{ij}는 0또는 1의 이항변수이므로

$$P(a_{ij} = 1) = E(a_{ij})$$

가 성립되고 이것은 시스템상태간의 유사관계를 고려한 다른 a_{ij}들의 가중 평균으로 산출될 수 있다. 즉, 동일한 조작이 문제의 시스템상태와 유사성이 높은 다른 시스템상태들에서 갖는 값을 이용한 추정치를 사용할 수 있다. 이를 표현하면

$$E(a_{ij}) = \sum_k w_{ik} a_{kj}$$

이 되고 적절한 가중치 w 의 정의가 문제로 남는다. 이 가중치의 정의는 곧 전체 시스템에 대한 사용자의 인식을 정량화하는 것이므로 지식상의 전개에 앞서 그에 요구되는 성질이 충분히 검토될 필요가 있다.

4.3 가중치를 통한 스키마 구조의 반영

위에서 논한 가중치 w 는 사용자가 시스템의 이해에서 사용하는 인지지도(Cognitive Map)상의 거리에 해당되는 수치로서 그 성질상의 중요한 요구사항들은 다음과 같이 정리된다.

1. 문제의 시스템상태(즉 S_i)에 높은 유사성을 갖는 시스템상태(즉 S_k)에 높은 가중치가 주어져야 한다.
2. 서로 매우 유사한 행태의 시스템상태들로 형성된 그룹의 존재할 때, 그 영향력은 그 그룹, 즉 스키마 단위로 안정되어야 하며 그에 속한 각개 시스템상태가 개별적으로 영향력을 갖는 것은 통제되어야 한다.
3. 가중치는 각 시스템상태들에 대한 인위적이거나 직관적인 판단을 배제한 방식으로 산정되어야 한다. 즉, 분석자에 의한 시스템상태들의 분류등의 일이 일어나지 않으면서도 위 1, 2의 조건이 지켜져야 한다.

첫번째 요구사항은 자명한 것이라 할 수 있다. 이것은 사용자가 사용하는 유비(Analogy), 또한 스키마 형성의 전략을 정량적으로 표현한 것이다. 한가지 여기서 지적해야 할 것은 이러한 사용자의 전략들은 인지적 복잡도를 낮추고자 하는 것이며, 이 요구사항이 준수되는 정량적 모형에서는 이러한 전략이 시스템의 엔트로피를 낮추는 데 기여하게 된다는 것이다.

두번째 요구사항은 설명을 필요로 한다. 어떤 시스템의 설계단계에서 사소한 차이의 시스템상태들이 분화될 가능성이 있는데 이렇게 발생한 하나의 시스템상태의 수적 증가가 a_{ij} 의 추정에 영향을 미치는 것은 최소화되어야 한다. 물론 다른 시스템상태들과 매우 다른 행태의 독립적인 시스템상태가 발생했을 때는 예외이다. 예를들어 A, B 두가지 유형의 시스템상태 그룹이 형성되어 각기 상반된 영향을 a_{ij} 의 추정에 미치고 있다 하자. 이때 B유형에 속한 시스템상태 수가 A유형에 속한 시스템상태 수보다 많다든가 하는 것이 추정치를 좌우하지 않아야 함은 당연하다. 이것은 사용자가 각 개체가 아닌 스키마적 지식구조를 활용함을 의미할 때 자연스러운 조건으로 이해될 수 있다.

세번째 요구사항은 분석방법의 객관성 확보라는 면에서 필연적인 것이지만, 전술한 두번째 요구사항의 성취를 어렵게 한다. 두번째의 요구에 대한 가장 자명한 대안은 곧 시스템상태들이 형성하는 스키마적 그룹을 파악하여 그 그룹에 대한 가중치를 정의하는 것이지만 이와 같은 분석자의 주관적 판단의 개입은 객관적 평가기법을 개발하려는 본 연구의 목적에 위배된다.

결국 본 논문의 앞부분에서 논한 사용자의 스키마적 지식구조는 이 가중치의 정의에서 구현되어야 하는데, 이를 위한 가장 기초적 작업은 당연히 시스템상태간의 유사지수 산정이다. 두 시스템상태간의 유사성은 어떤 조작이 그들간에 공통적으로 가능 또는 불가능한가에 의해 결정되는데, 각 조작이 균등하게 고려되기 보다는 그 가능성의 조합이 갖는 정보량에 의해 가중되는 것이 합당하다. 이러한 가중방식은 문서의 자동분류기법에도 적용되어 그 합리성이 검증된 바 있다[6].

이제 각 시스템상태가 자신과 같은 스키마에 속한 상태의 수에 반비례하여 갖는 가중치 α_i ($i=1, \dots, m$)을 가정하면 한 조작 O_j 가 어떤 상태에서 가능할 확률과 불가능할 확률은 각각

$$P(a_{\cdot j}=1) = \sum_i \alpha_i \cdot a_{ij} = \bar{a}_{\cdot j}$$

$$P(a_{\cdot j}=0) = 1 - \bar{a}_{\cdot j}$$

이 되고, 상태 i 와 상태 k 가 독립적이라는 가정 아래 조작 j 의 실현치가 갖는 정보량은

$$I_{ik}^j = \log\{P(a_{\cdot j}=a_{ij}) \cdot P(a_{\cdot j}=a_{kj})\}^{-1} \\ = -\log P(a_{\cdot j}=a_{ij}) - \log P(a_{\cdot j}=a_{kj})$$

이 된다. 이제 시스템상태간의 유사지수의 행렬

$$C = \{c_{ik} \mid i = 1, \dots, n; k = 1, \dots, n\}$$

$$c_{ik} = \sum_j S_{ik} I_{ik}^j / \sum_j I_{ik}^j, \text{ where } S_{ik} = \{j \mid a_{ij} = a_{kj}\}$$

로 정의되며 $c_{ik} \sim [0, 1]$ 이다.

예를 들어 위의 <표 2>에 나타난 시스템상태들 간에는 <표 3>과 같은 유사지수 행렬이 생성된다.

<표 3>은 S_2 와 S_3 , S_4 사이의 강한 유사성을 보여주고 있다. 이것은 곧 사용자가 이 세 시

스템상태를 하나로 묶는 스키마를 형성할 수 있음을 암시한다.

이제 우리가 a_{21} 의 기대치를 추정하려 한다면 a_{31} 과 a_{41} 이 유사성이 적은 시스템상태로 부터의 a_{11} 보다 큰 가중치를 지녀야 함을 알 수 있다. 그러므로 시스템상태 i 의 요소들을 추정하는데 사용될 시스템상태 j 의 가중치를

$$w_{ij} = c_{ij} / \sum_l c_{il}$$

와 같이 간략히 구할 수 있는 것으로 보인다. 그러나 이 가중치는 위의 두번째 요구사항에 위배되는 특성을 가지고 있다. 즉 새로운 시스템상태가 추가로 분류될 때, 그 추가되는 시스템상태와 같은 스키마에 속하지 않는 다른 시스템상태들의 영향력이 감소하게 되며 그것이 속한 스키마의 영향력은 정당한 이유없이 커지게 된다. 서로 강한 유사관계에 있는 시스템상태들의 가중치의 합계는 그러한 시스템상태의 수에 대하여 둔감하여야 하며, 따라서 이는 다른 많은 시스템상태들과 강한 유사관계에 있는 시스템상태의 영향력이 그만큼 하향조정됨으로써 달성될 수 있다. 따라서 위의 유사지수 행

<표 3>

<표 2>로 부터의 유사지수 행렬 C

	S_1	S_2	S_3	S_4
S_1	1	0.082	0	0.281
S_2	0.082	1	0.650	0.624
S_3	0	0.650	1	0.361
S_4	0.281	0.624	0.361	1

<표 4> 극단적 Cluster가 형성된 경우의 유사지수행렬

	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆
S ₁	1	1	.5	0	0	0
S ₂	1	1	.5	0	0	0
S ₃	.5	.5	1	.7	.7	.7
S ₄	0	0	.7	1	1	1
S ₅	0	0	.7	1	1	1
S ₆	0	0	.7	1	1	1

렬 C로부터

$$c'_{ij} = c_{ij} / \sum_{k \neq i} c_{kj} \quad \text{for } i \neq j$$

$$c'_{ij} = 1 \quad \text{for } i=j$$

를 구하면 이는 시스템상태간의 Cluster현상이 감안된 각 시스템상태의 유사성 분배의 정도가 된다. $\sum_{k \neq j} c'_{kj} = 1$ 이 성립하므로 그런 의미에서 이를 정규화 유사지수라고 할 수 있다 이제 다시 가중치는

$$w_{ij} = c'_{ij} / \sum_i c'_{ij}$$

로 정의하면 이것은 위의 요구사항 3이 준수된, 즉 주관이 배제된 산정방식이면서도 요구사항 1과 2를 만족하게 된다.

이 가중치의 성질을 검토하기 위하여 극단적인 예로 <표 4>와 같은 유사지수 행렬을 검토해 보자

여기에서 우리는 a₃₁을 추정하기 위한 w₃ vector를 구하려 한다. 먼저 S₆가 없을 때의 c₃ 벡터는 (.25, .25, 0, .35, .35)가 된다. 그리고 S₆가 추가된 때의 c₃ 벡터는 (.25, .25, 0, .233, .233, .233)가 되어 뒤의 세 시스템상태로 이루어진 Cluster의 전체 영향력은 일정하게 유지된 것을 볼 수 있다. 또 두 Cluster사이의 영

향력 배분이 여전히 0.5 대 0.7로 이루어진 것도 관찰할 수 있다.

4.4 엔트로피 산출의 예

이제 <표 2>의 예로 돌아가서 시스템상태 4에서 각 조작이 가능한 확률을 추정하기 위하여 c₄' 벡터를 구해보면

$$c'_4 = (0.281/1.082, 0.6241/1.733, 0.361/1.650, 1./1.266)$$

이 되고 이들의 총합은 0.911, 그리고 이것으로 c₄'을 나누어 얻어지는 가중치의 벡터 w₄는

$$w_4 = (.183, .488, .329, 0)$$

이 되어 a₄₁의 경우 그 추정치는 0.817이 된다. 이것은 곧 a₄₁이 1이 될 확률인데, 실제로 그 값이 1로 나타났으므로 이에 관련된 정보량은

$$I(a_{41}) = \log(1/0.817) = 0.292$$

로 산출된다. 이 작업을 표 2의 각 요소에 대하여 시행하여 그 총합을 구하고 그것을 요소의 수로 나누면 시스템 전체의 엔트로피가 산출된다.

이와같이 산정된 엔트로피는 곧 사용자가 한 시스템상태에서 어떠한 조작들이 가능 또는 불가능하다는 지식을 습득하는데 소화해야 하는 평균 정보량이 되며, 그 총합은 해당 인터페이스의 인지적 복잡도에 대한 직접적인 측정치로 활용될 수 있다. 어떤 인터페이스의 개념설계에서 조작기능의 집합이 주어졌을 때, 그에 대한 상이한 배치의 구조적 복잡도는 이 엔트로피 추정치를 사용하여 쉽게 상대적인 비교를 할 수 있다. 일례로 새로운 조작기능의 추가없이 동작방식등을 변형 또는 개선하려는 대안이 사용자의 인지적 복잡도를 감소시키는지 증가시키는지의 여부를 판정하는 등의 목적에 유용할 것이다. 이러한 효용은 특히 시스템 개발기법으로 최근 대두되고 있는 Rapid Prototyping과정에서 매우 중요한 단계인 인터페이스 평가에 유효한 도구로 쓰일 수 있는 가능성을 제시하고 있다.

5. 결 론

현대적 시스템의 기능상의 제한은 정보전자 기술상의 문제에서 보다 사용자가 그 기능을 충분히 활용하지 못하게 되는 데에서 오는 경우가 점점증하고 있다. 따라서 사용자 인터페이스의 설계와 개선이 극히 중요하게 되며, 그 평가와 개선의 핵심은 복잡도의 효과적 감소를 모색하는데 있다. 복잡도의 감소는 일관성을 통하여 도모할 수 있게 되므로 스키마의 개념을 도입한 시스템 분석방법은 복잡도의 평가에 필수적이며, 이러한 사용자의 지식구조를 반영하여 객관적으로 복잡도를 평가할 수 있는 방법이 요구된다.

본 논문은 사용자가 시스템 내에서 계획수립을 수행하는 데 필요한 지식의 복잡도를 정량적으로

분석하는 방법론을 개발하였으며, 특히 평가자의 주관적 판단에 의존하지 않고 인간의 스키마적 지식구조를 반영하여 정량적으로 분석하는 방법론을 제시한 데에 의의가 있다 할 수 있다.

본 연구는 사용자지식에 대한 인지공학적인 원리에 근거를 두고 있으나, 장래에 보다 정밀한 논의와 확장을 위하여는 실험적인 결과의 뒷받침도 있어야 할 것으로 사료된다. 단 실험적 검증은 연구의 영역화를 수반할 개연성이 큰 관계로 본 논문의 범위에서는 일단 제외하였고 후속 연구의 주된 과제가 되리라 생각한다. 본 연구의 방법론은 비영역적이며 일반적인 것이므로 소프트웨어 인터페이스, 가전제품 인터페이스, 의사결정 시스템, 기타 인간기계 체계에 광범위한 적용이 가능할 것으로 사료된다.

참 고 문 헌

- [1] Anderson, J.R., *Cognitive Psychology and Its Implications*, Freeman and Co., 1980.
- [2] Card, S.K., T.P. Moran and A. Newell, *The Psychology of Human-Computer Interaction*, LEA, 1983.
- [3] Helander, M.(Ed.), *Handbook of Human-Computer Interaction*, North-Holland, 1988.
- [4] Preece, J. and L. Keller (Ed.), *Human-Computer Interaction*, Prentice-Hall, 1990.
- [5] Rouse, W.B. and S.H. Rouse, *Measure of Complexity of Fault Diagnosis Tasks.*, IEEE SMC, 1979.
- [6] Salton, G. and M.J. McGill, *Introduction to Modern Informantion Retrieval*, McGraw-Hill, 1983.