

소프트웨어 제품을 위한 평가 선정 모형의 조사 및 적용성에 관한 연구

박 호 인[†] · 정 호 원^{††}

요 약

다양한 소프트웨어 제품의 급격한 증가로 인하여 소프트웨어 제품의 평가·선정을 위한 체계적이고 객관적인 방법이 필요하게 되었다. 본 연구에서는 소프트웨어 제품의 효과적 평가·선정을 위해 평가 과정의 핵심인 가중치 부여와 모형의 선정에 중점을 둔다. 이를 위해 첫째, 계층적 분석과정을 이용하여 평가 속성에 일관적 가중치를 부여한다. 둘째, 소프트웨어 제품의 선정 문제의 성격에 알맞은 모형을 조사·분류하여 모형별 장·단점을 분석한다. 적용 모형은 4개의 보상모형과 7개의 비보상모형으로 구성되어 있다. 선정된 모형은 특정 소프트웨어 제품(데이터베이스 모델러)에 응용되어 모형별로 제품을 평가한다. 본 연구는 가중치 부여 및 모형의 장 단점 분석과 응용 절차를 통하여 사용자의 다양한 요구사항에 대한 모형의 적용성을 제고하고자 한다.

A Study on Survey and Applicability of Evaluation and Selection Models for Software Products

Ho-In Park[†] · Ho-Won Jung^{††}

ABSTRACT

The rapid increase in the use of many commercial software products has necessitated a systematic and objective method of their evaluation and selection. Our study focuses on the assignment of weights and choice of proper models. First, the weights of attributes are assigned consistently by using the analytic hierarchy process. Second, many models, which can be suitable for the structure of evaluation and selection for software product, are collected, categorized into two types of model, and compared in terms of their strength and weakness. The models involved are four compensatory models and seven noncompensatory models. Finally, they are analyzed through the application of specific software products(database data modelers) in terms of their attributes. Our study enhances the applicability of models to a variety of user requirement utilizing the evaluating procedure and applications.

1. 서 론

개인용 컴퓨터의 확산과 최종사용자 컴퓨팅의 등장

으로 인하여 대부분 소프트웨어는 상용화된 다양한 제품 형태로 대량 생산·유통되고 있다. 이에 따라 사용자의 다양한 소프트웨어 제품에 대한 평가·선정 요구는 증대되고 있고, 이를 충족시켜줄 소프트웨어를 평가·선정할 수 있는 기술적 방법 및 표준은 제공되지 못하고 있다[1]. 한편 정보 시스템의 핵심 요소인 소프

† 정 회 원:부천전문대학 사무자동화과

†† 정 회 원:고려대학교 경영학과

논문접수:1996년 8월 13일, 심사완료:1997년 5월 12일

트웨어는 시스템에 융통성과 가치를 부여하는 반면 동시에 많은 문제의 근원이기도 하다. 따라서 정보 시스템의 기본인 소프트웨어 제품의 평가·선정은 효과적인 정보 시스템의 구축과 운영을 위해서는 필수적이다.

소프트웨어 제품은 다른 제품과 비교할 때 몇 가지 차이점을 가지고 있다. 첫째, 소프트웨어 제품 평가기준이 되는 속성들의 다양성과 무형성을 들 수 있다. 즉 소프트웨어 제품을 평가·선정함에 있어 사용자의 선호, 요구사항 등을 반영하기 위한 다양한 속성들이 존재할 수 있다. 또한 이러한 속성은 무형적이며 상당히 주관적일 수 있고 인지된 중요도도 상이할 수 있다는 것이다. 둘째, 다수의 평가 속성들간에 목표달성에 대한 상충성(conflictivity)이 존재한다는 것이다. 다수의 상충되는 목표들이 주어졌을 때, 유일한 최적해는 존재할 수 없으며, 다른 목표를 희생시키고 단일 목표를 개선시키는 절충(trade-off)이 필요해진다. 이러한 평가 속성의 특징으로 인하여 소프트웨어 제품의 평가·선정 문제가 구조상 다속성 의사결정(multiple attribute decision making)의 성격을 가지게 된다.

소프트웨어 제품의 평가·선정에 관한 기존 연구는 평가 속성에 관한 연구와 평가 모형의 적용에 관한 연구로 집약될 수 있다. 평가 속성에 관한 연구와 관련하여 Visker 등[2]은 중·소 규모의 기업에서 사용될 수 있는 소프트웨어 제품의 선정 기준으로서 소프트웨어 사용가능성, 개발 자원, 시간적 유효성, 재무적 요인, 조직과 정보시스템간의 적합성을 선별하였다. Lucas 등[3]이 제시한 소프트웨어 제품의 성공에 영향을 미치는 요인은 소프트웨어 제품의 속성, 사용자의 요구사항, 조직적 특성 등이 있다. Davis[4]는 유용성과 사용편의성으로 구성된 기술 수용 모형(technology acceptance model)을 소프트웨어 선정 요인으로 제시하였다.

평가 모형의 적용 분야에서 Sanders 등[5]은 유·무형 요인을 함께 고려한 Brown 등[6]의 공장입지모형을 적용하고, 평가 속성을 객관적 요인과 주관적 요인으로 나누어 재무·회계용 소프트웨어 제품을 평가하였다. Zahedi[7]는 다기준 의사결정 기법의 일종인 계층적 분석과정(Analytic Hierarchy Process: AHP)[8]을 이용하여 2개의 마이크로 컴퓨터용 DBMS 소프트웨어 제품을 평가하였다. 앞에서 언급한 기술 수용 모형이

Szajnal[9]에 의해 또 다른 DBMS 소프트웨어 제품에 적용되었다. 소프트웨어 품질 평가 분야에서 Jung 등[10]은 ISO/IEC 9126[11]이 제공하는 6개의 주요 소프트웨어 품질특성에 AHP를 적용하여 객관적인 품질 평가 모형을 제시했다. Anderson[12]은 “heuristic”이란 방법을 활용하여 워드프로세서, DBMS, 스프레드시트 프로그램 등의 소프트웨어 제품을 평가하였다.

상기의 서술된 각종 연구는 소프트웨어 제품의 평가에 상당한 기여를 하고 있다. 그러나 이러한 연구들은 일반적으로 한 모형만을 사용한 결과 단편적인 정보만을 제공함으로써 종합적이고 객관적인 정보의 제시에는 상당한 한계가 있다. 또한 평가기준이 되는 속성문제에서는 동일한 가중치를 가정하거나 임의적으로 가중치를 부여함으로써 최종 결과를 부정확하게 만드는 요인이 되고 있다. 이러한 관점을 바탕으로 본 연구에서는 기존의 소프트웨어 제품 평가·선정에 관한 연구에서 간과한 문제를 보완하고, AHP 및 모형의 장·단점 분석과 실제 적용을 통한 사용자에 대한 모형의 적용성을 향상시키는데 목적을 둔다.

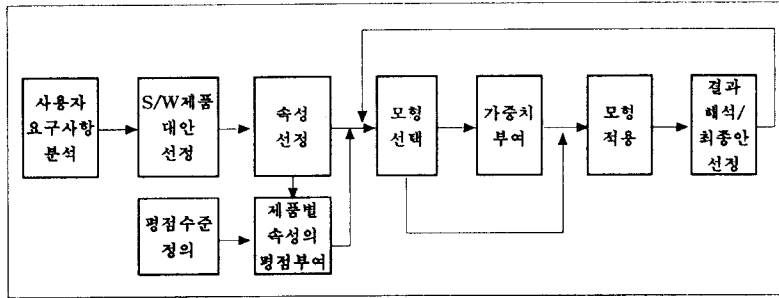
이후의 연구 내용은 2장에서 소프트웨어 제품에 대한 평가·선정의 문제를 개괄적으로 살펴보기 위해 전반적인 소프트웨어 평가 시스템을 제시한다. 3장에서는 사용자의 속성에 대한 가중치를 일관적으로 산정하기 위한 AHP를 설명하고, 4장은 소프트웨어 제품의 평가·선정에 이용될 수 있는 모형을 보상 모형과 비보상 모형으로 분류하고 장·단점을 비교한다. 5장에서는 평가 모형에 실제 자료를 적용한 결과를 종합한다. 6장은 평가 모형의 장 단점을 요약하고 결론을 맺는다.

2. 소프트웨어 평가 시스템

본 연구 내용과 전체적인 시스템과의 관련성을 파악하기 위하여 절차 중심의 소프트웨어 평가 시스템의 개관을 살펴보기로 한다.

2.1 소프트웨어 평가 시스템의 구성

소프트웨어 평가 시스템은 (그림 1)에서와 같이 종합적으로 나타낼 수 있다. 본 평가 시스템은 ISO[13]의 소프트웨어에 관한 평가절차와 일반적 의사결정 분석 절차를 조합하여 재구성한 것이다.



(그림 1) 소프트웨어 평가 시스템
(Fig. 1) Software evaluation system

2.2 평가 절차

평가 절차는 사용자의 요구사항 분석을 시작으로 하여 평가결과의 해석 및 최종안 선정의 단계로 완료되는 바, 평가결과에 따라 순환적 과정을 거치게 되는 특징을 가지고 있다. 세부적인 절차는 아래와 같다.

단계 1(사용자의 요구사항 분석): 일반적으로 소프트웨어 제품의 평가·선정의 과정은 사용자의 소프트웨어 제품에 대한 요구사항을 만족시키기 위한 것이다. 따라서 소프트웨어 평가 시스템은 사용자의 요구사항을 파악하는 것으로부터 시작한다.

단계 2(소프트웨어 제품 대안 선정): 사용자의 요구사항 분석이 완료되면 이러한 요구사항을 충족시킬 수 있는 적절한 대안의 집합을 구성한다. 일반적으로 사용자의 요구사항을 충족시킬 수 있는 대표적인 해당 소프트웨어 제품을 선정한다.

단계 3(속성의 선정): 평가 기준이 되는 소프트웨어 평가속성을 적절한 수준으로 선정한다. 이 절차는 평가대상이 되는 소프트웨어 제품의 특성과 성격에 따라 결정되고, 사용자의 요구사항에 따라 변경될 수 있다.

단계 4(평점수준의 정의): 요구사항의 만족 정도에 따라 제품별 속성에 대한 평가 범위를 나눈다(예를 들면 아주 좋음, 좋음, 보통, 나쁨 등). 평점수준에 대한 일반화는 불가능한 바, 평가 대상인 소프트웨어 제품의 특성에 따라 정의되어야 한다.

단계 5(제품별 속성의 평점 부여): 제품별로 주어진 평점수준에 따라 평점을 부여한다. 그러나 전문가가 아닌 사용자는 평가 대상이 되는 소프트웨어 제품군의 특성을 제대로 파악하기 어려워 평점 부여에 한계가 있다. 따라서 전문가나 평가기관에서 발표된 평가

결과를 인용하는 것이 바람직하다.

단계 6(모형의 선정): 사용자의 요구사항에 따라 알맞은 평가 모형을 선정한다.

단계 7(가중치의 부여): 사용자의 다양한 선호, 요구사항을 반영하기 위하여 가중치를 산정한다. 선정된 모형에 따라 가중치의 사용여부가 결정되고, 우선 순위가 필요할 경우도 있다.

단계 8(모형 적용): 선정된 모형에 대안의 평점 자료를 적용하여 평가결과를 산정한다.

단계 9(결과해석 및 최종안의 선정): 평가 결과를 해석하고 최종안을 선정한다. 최종안이 선정되지 못했거나 평가결과가 만족스럽지 못할 경우 단계 6으로 피드백된다.

3. 계층적 분석과정

소프트웨어 제품의 평가·선정의 문제는 매우 복잡하고 비구조적인 성격을 가지고 있는 바, 문제의 분석은 체계적인 접근방법을 필요로 한다. Saaty에 의해 제안된 AHP는 복잡한 문제를 작은 부분으로 나누어 문제를 단순화하고 체계화하는 의사결정방법이다[8, 14]. 특히 평가 기준이 되는 소프트웨어 각각의 속성에 가중치를 부여할 때 속성의 수가 많아짐에 따라 속성간의 가중치의 일관성 유지가 인간 인지력의 한계로 인하여 어려워지게 된다. 이러한 경우 AHP는 비교시점에서 비교대상을 두 개로 단순화시킴으로써 비교상의 복잡성과 비일관성을 보완할 수 있다.

3.1 AHP의 절차

AHP를 이용한 여러 속성간의 체계적인 가중치의 부여는 아래의 3단계로 구성되어 있다.

단계 1(문제의 계층적 표현):문제의 목표를 정의하고 목표에 영향을 미치는 관련 속성을 계층적으로 세분화하여 의사결정구조를 설정한다. 최상위 계층은 문제의 최종적인 목표를 나타내고 다음 계층은 이러한 목표에 영향을 미치는 속성을 표현한다. 그 다음 계층은 앞 계층에 영향을 미치는 세부속성을 나타낸다.

단계 2(쌍대비교(pairwise comparison)):단계 1에서 작성된 계층구조를 바탕으로 계층별로 쌍대비교를 시행한다. 사용자의 쌍대비교를 통해 행렬 $A = (a_{ij})$ 가 이루어지며, a_{ij} 는 w_i/w_j 의 추정치이고, $a_{ji} = 1/a_{ij}$ 로 표현된다. 행렬 A 는 주대각선의 원소들이 모두 1이 되는 대칭적 역수행렬(reciprocal matrix)이다. 이 행렬에 상대적 중요도를 나타내는 열 벡터(column vector) $W = (w_1, w_2, \dots, w_n)^T$ 을 곱한 결과는 $A \cdot W = n \cdot W$ 가 되고 행렬 A 를 알고 있다면 $(A - n \cdot I)W = 0$ 이 된다. n 은 행렬 A 의 고유값(eigenvalue)이고 벡터 W 는 고유 벡터(eigenvector)라 한다.

단계 3(근사계산법):가중치를 간편하게 구하기 위해 첫째, 단계 2에서 구한 행렬 A 를 열별로 합계를 구한다. 각 원소를 해당 열별 합계로 나누어 정규화를 시킨다. 둘째, 상기 결과를 행별로 평균을 구하면 해당 행별 평균이 구하고자 하는 가중치의 추정치가 된다. 마지막으로 행렬 A 와 가중치벡터 W 를 곱한 결과로 얻어진 벡터의 원소를 해당 가중치로 나눈 값을 평균 하면 추정된 최대 고유값(estimated largest eigenvalue: λ_{max})이 된다.

3.2 일관성의 측정

AHP에서 쌍대비교를 통해 행렬을 구성할 때 속성의 수가 많은 경우 인지력의 한계로 인하여 완벽한 논리적 일관성을 유지하는 것이 어렵다. 쌍대비교에서 가중치의 부여가 일치하지 않는다면 행렬 A 에서 불일치의 정도는 $\lambda_{max} - n$ 에 의해 측정될 수 있다. 이 값은 일관적인 추정치와 실제 쌍대비교 결과와의 판단상의 오차라고 할 수 있다. AHP는 $\lambda_{max} - n$ 을 이용하여 쌍대비교에서 불일치의 정도를 측정하는 도구로 아래와 같은 일관성 비율(Consistency Ratio:C.R.)을 제공하고 있다.

$$C.R. = \frac{C.I.}{R.I.} = \frac{\lambda_{max} - n}{n - 1} \cdot \frac{1}{R.I.}$$

상기 식에서 C.I.(Consistency Index)는 고유값에서 n (요소의 수)을 뺀 값을 $(n-1)$ 로 나눈 값이고 R.I.(Random Index)는 C.I.를 경험적 자료로 얻어진 평균 무작위 지표이다. 일관성 비율이 10%이내일 경우는 일반적으로 허용되지만 20%이상일 경우는 행렬 A 의 재구성을 필요로 한다[14].

4. 소프트웨어 평가 모형

소프트웨어 제품의 효율적 평가 및 선정을 위하여 대상 제품의 특성과 구조에 적합하도록 평가 모형을 선정해야 한다. 선정된 모형은 평점간의 절충 허용여부에 의해 크게 보상(compensatory) 모형과 비보상(non-compensatory) 모형으로 나눌 수 있다. 보상 모형은 속성간의 평점 교환이 허용되는 모형으로 한 개 이상의 속성에서 얻은 좋은 평점으로 그밖에 다른 한 개 이상의 속성에서 얻은 나쁜 평점을 상쇄시킬 수 있다. 반면에 이러한 속성 평점간의 절충이 허용되지 않은 모형을 비보상 모형이라고 한다. 적용 모형에서 공통적으로 사용되는 부호를 정의하면 아래와 같다.

- i : i 번째 소프트웨어 제품($i = 1, \dots, m$)
- j : j 번째 속성($j = 1, \dots, n$)
- r_{ij} : 소프트웨어 제품 i 의 속성 j 의 평점
- w_j : 속성 j 에 부과된 가중치

4.1 보상 모형

보상 모형은 각 속성간의 절충이 허용되는 모형으로 일반적으로 정량적 자료에 한하여 사용될 수 있다.

4.1.1 선형 가중 모형(linear weighted attribute model)

이 모형은 소프트웨어 제품의 평가에서 가장 많이 사용되는 모형으로 모든 속성과 사용자가 부여한 가중치를 함께 고려하여 최종안을 구한다[1]. 이 방법은 모형에 대한 이해와 사용이 용이하다는 장점이 있는 반면 속성들간에 상호선택독립(mutual preferential independence) 조건을 만족시킬 때에만 정당화될 수 있기 때문에 속성의 선정에 신중해야 한다. 상호선택

독립 조건은 대부분 보상 모형에서 충족되어야 한다. 본 모형의 적용단계는 다음과 같다.

단계 1: 각 소프트웨어 제품별 평가값 $Q_i = \sum_{j=1}^n w_j \cdot r_{ij}$ 로 구해진다.

단계 2: 각각의 평가값을 서수적으로 비교하여 최대 값을 갖는 대안 $Q^* = \max_i Q_i$ 이 최종안으로 결정한다.

4.1.2 선형 배정 모형(linear assignment model)

이 모형은 모든 속성과 모든 소프트웨어 제품을 동시에 고려하여 각 제품의 유일한 순위를 결정한다. 이 방법은 각 제품의 유일한 순위가 증시될 때 유용하게 적용될 수 있는 장점이 있다. 그러나 각 속성별 순위의 빈도만이 평가에 이용되기 때문에 나머지 다른 관련 정보는 모두 무시된다는 단점이 있다. 아래 과정을 통해 최적해에 이르게 된다[1].

단계 1: 개별 순위행렬 $C_{(m \times n)} = \{c_{ij}\}$ 를 구하고, 이 행렬을 이용하여 빈도행렬 $\Pi_{(m \times m)}$ 을 구한다($0 \leq \pi_{ij} \leq m$). 행렬 π_{ij} 는 소프트웨어 제품 i 가 순위 j 에 해당되는 빈도를 나타내고, 행렬 c_{ij} 는 소프트웨어 제품 i 에서 속성 j 의 순위를 나타낸다($1 \leq c_{ij} \leq m$).

단계 2: 종합 순위행렬 P 를 아래와 같이 정의한다.

$$P_{ij} = \begin{cases} 1, & \text{제품 } i \text{가 품질순위 } j \text{를 가지면,} \\ 0, & \text{그렇지 않으면.} \end{cases}$$

각 제품은 유일한 순위에만 부여되고, 각 순위는 한 제품에만 부여되기 때문에 $\sum_{i=1}^m P_{ij} = 1$ 그리고 $\sum_{j=1}^m P_{ij} = 1$ 와 같은 제약식을 갖는다.

단계 3: 상기 단계에서 구해진 수식을 바탕으로 가능한 해들(S_m) 중에서 제약조건을 만족하면서 Q 를 최대로 하는 행렬 P 를 선형계획법에 적용하여 구한다.

$$\begin{aligned} \max Q^* &= \sum_{i=1}^m \sum_{j=1}^m \pi_{ij} P_{ij} \\ \text{s.t.} & \\ \sum_{j=1}^m P_{ij} &= 1, \quad i = 1, 2, \dots, m \end{aligned}$$

$$\sum_{j=1}^m P_{ij} = 1, \quad j = 1, 2, \dots, m$$

$P_{ij} = 0$ 또는 1.

상기 절차는 가중치가 동일한 것으로 가정하였지만 상이한 가중치의 경우에는 정의 및 절차에 있어 조정이 필요하다. 첫째, 행렬 C 를 속성 j 의 소프트웨어 제품 i 의 모든 가능한 순위의 집합으로 재정의한다. 둘째, 요소 P'_{ij} 는 확률분포를 나타내며 집합 c_{ij} 에 포함된 각 순위의 확률을 포함한다. 셋째, 벡터 m 을 $e_1 = (1, 0, \dots, 0), \dots, e_m = (0, 0, \dots, 1)$ 과 같이 정의한다. 넷째, Π 를 $\Pi_{ik} = \sum_{j=1}^n w_j \cdot (e_k \cdot P'_{ij})$ 로 재정의하여 단계 3에 대체하여 계산한다[1].

4.1.3 가중 차이 모형(additive-difference model)

이 모형은 대안간의 평점을 쌍대비교하여 차이의 합계를 구하고, 그 결과를 비교하여 최종안을 선정하여 간다[15]. 장점으로는 쌍대비교를 통해 대안간의 우열 관계를 관찰하기에 적합하다는 것이다. 단점으로는 평가 결과는 선형 가중 모형과 같은 반면 평가 절차가 다소 번거롭다는 것이다. 평가 절차는 아래와 같다.

단계 1: 제품별 평점의 차이의 합계 $d_i = \sum_{j=1}^n w_j(r_{ij} - r_{i+1,j})$ 를 계산한다($i = 1, 2, \dots, m-1$). d_i 가 양수이면 i 번째 대안이 우세하고, 음수이면 $i+1$ 번째 대안이 우세하다.

단계 2: 차이의 합계가 큰 대안을 선정하고, 선정된 대안과 다음 대안을 비교하기 위해 단계 1의 과정을 반복한다. 쌍대비교에서 보다 나은 대안을 계속 선정함으로써 마지막 대안과 비교하여 최종으로 남는 대안을 최종안으로 선정한다.

4.1.4 타협 해법(compromise programming)

타협 해법은 이상적 지점(ideal point)으로부터의 거리개념에 기초한 모형으로 이상적 지점으로부터의 거리가 최소인 대안을 선정한다[16, 17]. 이 방법의 장점으로는 타협안에 이르게 된 이유를 이론적으로 설명이 가능하다는 것이다. 또한 p 값에 따라 다수의 평

가결과를 제공함으로써 사용자에게 합리적인 의사결정을 가능하게 한다. 그러나 p 값에 따라 최종안이 다를 경우 사용자의 판단에 혼란을 야기시킬 수 있다. 다음 단계를 통해 타협안이 구해진다.

단계 1: 평점행렬로부터 이상적 지점 x_j^* , 반이상적인 지점(anti-ideal point) x_{j_0} 를 아래와 같이 구한다.

$$x_j^* = \max_i r_{ij} \quad i=1, 2, \dots, m, \quad j=1, 2, \dots, n$$

$$x_{j_0} = \min_i r_{ij} \quad i=1, 2, \dots, m, \quad j=1, 2, \dots, n$$

단계 2: p 의 관점에서 계산된 소프트웨어 제품 i 의 거리측정치 d_i^p 를 구한다. p 는 계량적으로 비교를 가능하게 하는 거리 차원이며 1에서 ∞ 까지 사용될 수 있다.

$$d_i^p = \left\{ \sum_{j=1}^n w_j^p \left(\frac{x_j^* - r_{ij}}{x_j^* - x_{j_0}} \right)^p \right\}^{1/p}, \quad i=1, \dots, m, \quad j=1, \dots, n.$$

p 값이 무한대인 경우는 아래와 같이 단순화된 공식을 이용한다.

$$d_i^\infty = \max_j \left\{ w_j \left(\frac{x_j^* - r_{ij}}{x_j^* - x_{j_0}} \right) \right\}, \quad i=1, \dots, m, \quad j=1, \dots, n.$$

일반적으로 p 값이 1, 2, ∞ 경우 기하학적으로 중요한 의미를 지닌다. $p=1$ 일 때 두점간의 거리는 모든 점을 포함하기 때문에 가장 큰 d 값을 가지게 된다. $p=2$ 일 때 두점간의 거리는 직선이기 때문에 가장 작은 차이를 보인다. 무한대의 경우는 가장 큰 차이를 보이는 속성의 평점이 d 값을 결정짓는다[16]. 특히 $p=\infty$ 인 경우 한 개의 속성이 d 값을 결정짓기 때문에 비보상적 성격으로 전환되게 된다.

단계 3: p 의 관점에서 최소의 거리를 갖는 대안을 타협해로 채택한다.

$$d_i^* = \min_i d_i^p, \quad i=1, \dots, m.$$

4.2 비보상 모형

이 모형은 사용자의 요구사항을 반영하기 위해 최소 허용 기준을 정하여 평가하는 기술적 모형(descrip-

tive model)이 주류를 이룬다.

4.2.1 지배 모형(dominance model)

이 모형은 대안의 초기 비교과정에서 주로 이용되고 있다. 특정 대안을 나머지 다른 대안 중 한 대안이 상과 비교하여 모든 속성에서 열등하거나 같다면 이 대안은 지배당하고(dominated)있다고 할 수 있다. 지배당하는 대안을 기존의 대안 집합에서 제외함으로써 비열등해(nondominated solution) 집합을 구한다. 축소된 대안은 그 밖의 다른 모형에 적용되어 최종안이 구해진다[18]. 이 모형은 열등한 대안의 제거를 통해 대안집합을 단순화할 때 유용하게 사용될 수 있는 반면 모형자체로는 최종안의 선정이 불가능하다.

4.2.2 최대최대 모형(maximax model)

이 모형은 가장 간단하고 쉬운 비보상 방법으로 모든 속성은 동일한 가중치를 가지고 있다고 가정하고 있다. 이 모형은 일반적으로 속성 평점의 변이가 적을 때, 속성의 중요도가 거의 같을 때, 비교적 낮은 하나의 속성이 사용자 만족에 심각한 영향을 주지 않을 때, 속성간의 공분산이 높을 때 알맞은 방법이다. 이러한 조건에서는 부가적인 정보의 가치는 낮고, 따라서 허술한 제품을 선정할 가능성은 낮아진다[1]. 그러나 사용 조건이 까다롭고 관련 정보의 활용도가 매우 낮다. 또한 사용자의 다양한 선호를 전혀 반영하지 못하는 단점이 있다. 다음의 과정을 통해 최종안에 이른다.

단계 1: 각 제품에 대한 최대 속성 평점 $Q_i^* = \max_j r_{ij}$ 을 구한다.

단계 2: 최대의 평점을 갖는 소프트웨어 제품 $Q^* = \max_i Q_i^*$ 을 선정한다.

4.2.3 사전식 순위법(lexicographic ordering)

이 모형은 속성의 우선 순위에 따라 순차적으로 한번에 하나의 목표만을 최대화하는 방법이다[18]. 이 모형의 장점으로 한, 두개의 속성만이 중요할 때 신속하게 대안을 선정할 수 있다는 것이다. 반면 단점으로 최소의 허용 기준이 없기 때문에 사용자의 선호에 대한 많은 정보를 반영하고 있지 못하고, 평가과정 초기에 최종안이 선정되면 해당 대안이 부실할 가

능성이 높다. 아래와 같은 절차로 진행된다.

단계 1: 중요도에 따라 속성의 순위를 부여한다.

단계 2: 가장 중요한 속성의 평점을 비교하여 평점의 크기에 의해 제품의 순위를 부여한다. 최고 순위에 유일한 제품이 있으면 그 제품이 최종안으로 선정된다. 두 개이상의 제품이 최고 순위에 속하면 다음 중요한 속성을 동순위가 깨지거나 모든 속성이 반영될 때까지 계속한다.

4.2.4 Elimination By Aspects(EBA)

Tversky[19]에 의해 제안된 EBA는 모든 속성에 최소 허용기준을 설정하고, 중요도에 따라 차례로 대안의 평점과 허용기준을 비교함으로써 최종안을 탐색하는 것이다. 이 모형은 적용이 쉽고, 계산과정이 거의 없으며, 우선 순위를 갖는 속성에 따라 탐색함으로써 평가 결과에 대해 정당화하기 쉽다는 장점을 갖는다. 단점으로는 평가 초기에 최종적 선정이 이루어질 가능성이 있다. 따라서 관련 정보의 상당 부분이 무시되기 때문에 선정된 제품이 이후의 속성에서 최소한의 평점도 얻지 못할 수 있다. 다음과 같은 단계를 통해 진행된다.

단계 1: 속성의 중요도에 의해 우선 순위를 정한다.

단계 2: 각 속성의 최소 허용기준을 부여한다.

단계 3: 가장 중요한 속성부터 최소 기준이 만족되는 제품을 선정한다. 선정된 제품군이 한 제품으로 구성되어 있다면 해당 제품이 최종안으로 선정된다. 2개이상의 제품이 있고 고려되지 않은 속성이 포함되어 있다면, 다음으로 중요한 속성을 평가한다. 모든 속성이 고려되었다면 평가 절차는 종료된다.

4.2.5 접속 모형(conjunctive model)

이 모형은 모든 속성에 대해 허용기준을 설정하여 허용기준과 대안을 비교하여 대안을 축소해 나간다 [20]. 이 모형의 장점은 모든 속성의 최소 기준을 만족해야 하기 때문에 최소 기준을 만족하는 대안의 선정이 가능하다는 것이고, 단점은 최소 허용 기준을 엄격하게 적용할 경우 모든 허용기준을 만족하는 대안이 없을 수도 있다는 것이다. 최종안의 선정은 아래와 같이 이루어진다.

단계 1: 모든 속성에 최소 허용기준을 설정한다.

단계 2: 허용기준에 미달하는 대안을 제거하면서 모든 속성의 허용기준을 만족하는 대안을 탐색하여 간다. 이러한 절차를 통해 모든 허용기준을 만족하는 대안이 최종적인 대안으로 채택된다.

4.2.6 비접속 모형(disjunctive model)

접속 모형이 모든 속성에 대하여 허용기준이 정해지는 반면 이 모형은 한 개이상의 속성에만 최소한의 허용기준을 지정한다. 대안이 몇몇 속성의 허용기준만 만족된다면 선정되는 것이다. 따라서 다른 비보상 모형이 열등한 대안의 신속한 제외에 그 목적이 있는 반면 이 모형은 대안의 신속한 채택에 그 목적이 있다. 지배 모형과 같이 최종적인 고려대상의 범위를 축소하기 위한 사전평가에 주로 이용된다[20]. 소수의 속성에서 일정한 기준이상을 필수적으로 만족해야만 하는 상황에는 유용하게 활용될 수 있다. 반면 다수의 대안이 선정될 수 있어 다른 모형과 함께 사용되어야 정확한 최종안의 선정이 가능하다. 다음의 단계로 구성된다.

단계 1: 중요하다고 판단되는 한 개이상의 속성을 선정하여 최소 허용기준을 부여한다.

단계 2: 중요도에 따라 허용기준을 비교하면서 만족하는 대안을 탐색하여 간다. 마지막으로 남는 대안이 최종안이 된다.

4.2.7 휴리스틱 모형(heuristic model)

이 모형은 여러 차원을 동시에 고려하여 대안을 평가하는 방법으로 Bernard Roy의 ELECTRE(ELimination Et Choix Traduisant la REalite) 등과 같은 유사한 모형이 다수 존재한다. 본 연구에서는 Anderson이 사용한 "heuristic"[12, 21]이라는 모형을 중심으로 기술한다. 휴리스틱 모형은 3가지 차원인 일치도(concordance), 불일치도(discordance), 그리고 중요도(magnitude)의 의해 평가된다. 이 모형의 장점은 사용자의 요구사항을 만족, 불만족 등 3개의 차원에서 종합적인 평가를 함으로써 관련 정보의 활용도를 극대화한다는 것이고, 단점은 계산절차가 약간 복잡하고 허용 기준의 부여가 까다롭다는 것이다. 아래와 같이 모형에서 사용되는 부호와 단계로 구성된다.

$$S_{ij}: S_{ij} = \{k | r_{ik} \geq r_{jk} \quad k=1, 2, \dots, n\} : \text{제품 } i \text{가 받은 평점이 제품 } j \text{가 받은 평점보다 크거나 같은 속성의 빈도에 대한 집합, } i=1, 2, \dots, m, j=1, 2, \dots, m.$$

$$d_{ij}: d_{ij} = \min\{r_{ik} - r_{jk} | r_{ik} \leq r_{jk}, k=1, 2, \dots, n\}.$$

$$Z: (\max_i \max_j r_{ij} - \min_i \min_j r_{ij}), i=1, 2, \dots, m, j=1, 2, \dots, n.$$

단계 1: 다음과 같이 제품의 만족도를 나타내는 일치도 행렬 S , 불만족도를 표시하는 불일치도 행렬 D , 그리고 빈도와 차이로 구성된 중요도 행렬 G 를 구한다.

$$S_{(m \times m)} = \{S_{ij}\}; S_{ij} = \frac{\sum_{k \in S_{ij}} w_k}{\sum_{k=1}^n w_k} \quad i=1, \dots, m, j=1, 2, \dots, m.$$

$$D_{(m \times m)} = \{D_{ij}\}; D_{ij} = \frac{|d_{ij}|}{Z} \quad i=1, \dots, m, j=1, 2, \dots, m.$$

$$G_{(m \times m)} = \{G_{ij}\}; G_{ij} = \frac{\sum_{k \in S_{ij}} (r_{ik} - r_{jk})}{|S_{ij}|} \quad i=1, \dots, m, j=1, \dots, m, k=1, \dots, n.$$

단계 2: 상기 3개의 행렬로부터 허용기준과 비교된 행렬 P 를 구한다.

$$P_{(m \times m)} = \{P_{ij}\}; P_{ij} = \begin{cases} 1, & \text{만일 } S_{ij} \geq T_s, G_{ij} \geq T_g, \text{ 그리고 } D_{ij} \leq T_d \text{ 이라면,;} \\ 0, & \text{그렇지 않으면, } i=1, \dots, m, j=1, \dots, m \end{cases}$$

만일 S_{ij} 와 G_{ij} 가 T_s 와 T_g 보다 크거나 같고, D_{ij} 가 T_d 보다 작거나 같다면, 이것은 소프트웨어 제품 i 가 소프트웨어 제품 j 보다 행렬 P, G 와 D 관점에서 좋다는 의미이다.

단계 3: 행렬 P 로부터 Kendall 점수($K_i = \sum P_{ij}$)를 계산한다. 점수의 순서에 의해 가장 높은 점수를 얻은 대안이 최고의 대안이 된다. 만일 특정 순위에 동순위가 있지 않다면 평가 절차는 끝나게 되고, 동순위가 존재한다면 단계 4로 넘어간다.

단계 4: 동순위가 있다면 해당 순위를 대상으로 P_{ij}

의 하위행렬을 다시 계산한다. 하위행렬은 동순위가 존재하는 소프트웨어 제품의 행과 열에 의해 구해진다. 재 계산된 Kendall 점수를 갖고 동순위 제품에 대한 순위를 다시 정한다. 동순위가 없다면 평가절차가 완료된다. 만일 그래도 동순위가 존재한다면 임의적으로 순위를 정할 수 있다.

4.3 모형의 비교

이상과 같이 소프트웨어 제품의 평가·선정에 적합한 모형을 보상·비보상 모형으로 분류하여 장·단점 및 절차를 정리하였다. 모형별 장·단점은 크게는 보

〈표 1〉 모형별 장·단점 비교
(Table 1) Comparison between models in term of strength and weakness

모형	장·단점	장 점	단 점
보상 모형	선형 가중 모형	모형에 대한 이해와 적용이 용이함	모형을 정당화하기 위해 속성의 선정에 신중해야함
	선형 배정 모형	유일한 순위의 부여가 가능함	관련 정보의 활용율이 저조함
	가중 차이 모형	대안간의 비교가 용이함	결과는 선형 가중 모형과 같은 반면 평가절차가 번거로움
	타협 해법	관련 정보의 활용도가 높고 비교 가능한 복수의 차원을 제공함	복수의 최종안의 존재로 인한 혼란의 가능성이 있음
비보상 모형	지배 모형	열등한 대안의 제거로 대안집합의 단순화가 가능함	최종안의 탐색이 불가능함
	최대최대 모형	대안의 선정이 신속함	사용 조건이 까다롭고 사용자의 선호도가 무시됨
	사전식 순위법	소수의 속성만이 중요할 때 신속한 결정이 가능함	허용기준이 없어 관련 정보의 활용이 저조함
	EBA	평가결과에 대한 이론적 정당화가 용이함	초기에 대안이 선정되면 관련 정보의 활용도가 감소할 수 있음
	접속 모형	최소한 기준을 갖는 대안의 선정 가능성이 높음	모든 허용기준을 만족하는 대안이 없을 수 있음
	비접속 모형	소수의 속성의 만족이 필수적인 상황에 알맞음	다수의 최종안이 선정될 가능성이 높음
휴리스틱 모형	다양한 차원의 종합적인 평가가 가능함	계산 절차가 복잡하고 허용 기준의 부여가 까다로움	

상·비보상 모형의 특징을 포함하면서 모형자체의 고유한 특징이 추가되어 나타난다.

보상 모형은 일반적으로 대안이나 속성의 수가 한정적인 때에 알맞고, 비보상 모형은 대안의 수가 다수인 경우 대안의 축소에 적절하다. 또한 보상 모형은 1회 시행으로 평가가 완료되는 반면 비보상 모형은 대체로 순환적 과정을 통해 최종안에 이르게 되는 특징을 가진다. 보상 모형은 정량적 자료에 한하여 분석이 가능하고, 대부분의 비보상 모형은 정량적 자료뿐만 아니라 정성적 자료에도 이용이 가능하다.

특히 비보상 모형 중 EBA, 집속 모형, 비집속 모형은 제품간 속성의 비교보다는 제품과 허용기준의 비교를 통해 평가가 진행되는 특징을 갖고 있다. 따라서 허용기준의 정도에 따라 최종안이 존재하지 않을 수도 있고, 반대로 복수의 최종안이 발생할 수도 있는 바, 최종안이 존재하지 않는다면, 단일 대안이 선정될 수 있도록 허용기준을 완화하여야 한다. 반대로 다수의 최종안이 존재할 경우 허용기준을 강화하거나 다른 모형을 적용하여 단일 대안을 유도하여야 한다. 그러므로 상기의 모형에서는 허용 기준이 매우 중요한 요소로 작용되므로 신중한 허용기준의 설정이 필수적이다. 지금까지 기술한 모형의 장·단점을 비교한 결과를 요약하면 <표 1>과 같다.

5. 평가 모형의 적용

모형의 적용은 실제 자료를 통한 평가 결과를 바탕으로 모형간 비교 및 적용상의 시사점 등을 파악하여 사용자에게 대한 모형의 적용성을 향상하기 위한 것이다.

상기에 열거된 모형의 응용을 위하여 Infoworld(Feb. 5, 1996)에서 발매된 컴퓨터 관련 제품의 비교 난에 수록된 데이터베이스 모델러 소프트웨어 제품의 평점 자료를 사용한다. 평가 대상 제품은 Bachman's Solution, Erwin ERX 2.5, S-Designor Enterprise 4.2.1, 그리고 Vivid Clarity 1.1.이다. 평가 점수는 10점 척도법에 의해 10.0 (excellent), 7.5 (very good), 6.25 (good), 5.0 (satisfactory), 2.5 (poor), 그리고 0.0 (unacceptable)이다. 각 속성에 따라 4개의 제품에 대한 평점은 아래의 <표 2>와 같다.

평가 속성은 성능과 지원 및 가격으로 구성된다. 성능에 대한 하위 속성은 논리 데이터모형 개발, 논

<표 2> 평점 행렬
<Table 2> Rating matrix

소프트웨어 제품 속성	Bachman's Solution	Erwin ERX2.5	S-Designor Enterprise 4.2.1	Vivid Clarity 1.1
성능				
논리 데이터모형 개발(DLDM)	6.25	7.50	6.25	5.00
논리 데이터모형 레포팅(RLDM)	7.50	6.25	6.25	2.50
물리적 데이터모형 작성(GPDM)	6.25	6.25	5.00	2.50
물리적 데이터모형 레포팅(RPDM)	7.50	6.25	6.25	2.50
데이터베이스 스키마 작성(GDS)	7.50	6.25	5.00	2.50
데이터베이스 유지보수(DM)	5.00	5.00	5.00	5.00
지원 및 가격				
문서화(DOC)	6.25	5.00	5.00	2.50
지원(SUP)	6.25	6.25	5.00	2.50
가격(PRC)	0.00	7.50	6.25	10.00

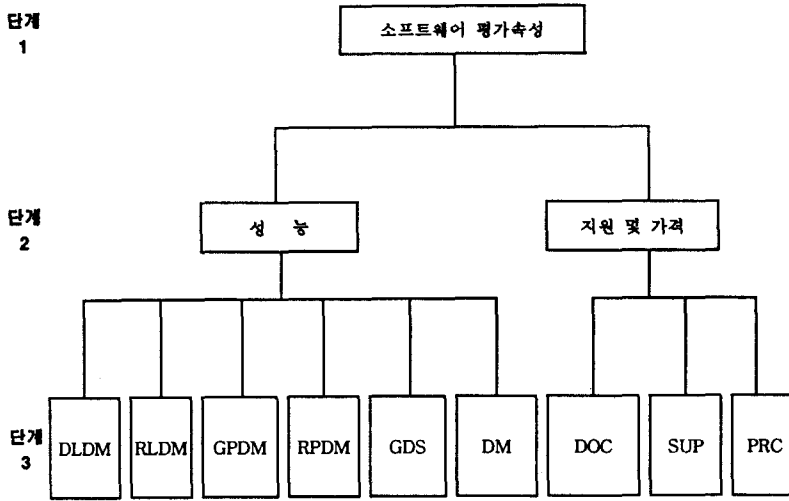
출처 : Infoworld(Feb. 5, 1996) pp60-65.

리 데이터모형 레포팅, 물리적 데이터모형 작성, 물리적 데이터모형 레포팅, 데이터베이스 스키마 작성, 그리고 데이터베이스 유지보수로 구성되어 있다. 지원 및 가격의 하위속성은 문서화, 지원, 그리고 가격으로 구성된다.

5.1 AHP에 의한 가중치의 부여

평가 모형의 적용을 위하여 Infoworld(Feb. 5, 1996) 자료를 바탕으로 하여 소프트웨어 평가속성의 계층적 구조도를 작성하면 (그림 2)와 같다. 계층적 구조도는 3단계로 구성되는 바, 최상위 단계는 궁극적인 속성인 소프트웨어 평가속성이 위치하고 다음 단계는 2가지 하위 속성인 그룹 1(성능)과 그룹 2(지원 및 가격)로 세분화된다. 그룹 1은 성능 속성에 영향을 미치는 6개의 세부 속성으로 구성되어 있으며 그룹 2는 지원 및 가격 속성에 영향을 미치는 3가지 세부 속성으로 구성되어 있다.

다음 단계로 속성 및 세부속성에 대한 가중치, λ_{max} ,



(그림 2) 평가 속성의 계층적 구조도
(Fig. 2) Hierarchical structure of evaluating attributes

그리고 일관성 비율을 AHP를 활용하여 <표 3, 4, 5>에서와 같이 구할 수 있다. 단계 3의 그룹 1의 일관성 비율이 0.007로 나타난 바, 이는 세부속성간의 쌍대비교가 매우 일관적으로 이루어졌다는 것을 의미한다.

<표 3, 4, 5>로부터 하위속성의 전체적인 가중치를 구하면 <표 6>과 같다. 전체적인 가중치는 $\sum w_j = 1$ 이

<표 3> 단계 2의 쌍대비교 행렬
<Table 3> Pairwise comparison matrix for leve 2

	성능	지원 및 가격	가중치
성능	1.00	3.00	0.75
지원 및 가격	0.33	1.00	0.25

$\lambda_{max} = 2.00$ C.R. = 0.00

<표 4> 단계 3에서 그룹 1의 쌍대비교 행렬
<Table 4> Pairwise comparison matrix for group1 in level 3

	DLDM	RLDM	GPDM	RPDM	GDS	DM	가중치
DLDM	1.00	5.00	1.00	5.00	1.00	1.00	0.25
RLDM	0.20	1.00	0.33	1.00	0.33	0.33	0.06
GPDM	1.00	3.00	1.00	3.00	1.00	1.00	0.21
RPDM	0.20	1.00	0.33	1.00	0.33	0.33	0.06
GDS	1.00	3.00	1.00	3.00	1.00	1.00	0.21
DM	1.00	3.00	1.00	3.00	1.00	1.00	0.21

$\lambda_{max} = 6.04$ C.R. = 0.007

<표 5> 단계 3에서 그룹 2의 쌍대비교 행렬
<Table 5> Pairwise comparison matrix group 2 in level 3

	문서화	지원	가격	가중치
문서화	1.00	3.00	3.00	0.60
지원	0.33	1.00	1.00	0.20
가격	0.33	1.00	1.00	0.20

$\lambda_{max} = 3.00$ C.R. = 0.00

<표 6> j번째 속성에 부여된 가중치
<Table 6> The weight w_j assigned to the jth attribute.

속성	DLDM	RLDM	GPDM	RPDM	GDS	DM	DOC	SUP	PRC
가중치	0.17	0.05	0.16	0.05	0.16	0.16	0.15	0.05	0.05

되도록 약간 조정한 것이다. AHP의 의해 구해진 가중치가 Infoworld가 제시한 가중치와 거의 차이가 없는 것으로 나타났다.

5.2 모형 적용상의 비교

선정된 모형을 적용한 결과를 요약하면 <표 7>과 같다. 모형을 적용함에 있어 특정한 조건이나 허용 기준이 필요한 모형의 경우 이를 비교란과 하단에 별도로 제시하였다.

<표 7>은 Erwin이 7개의 모형에서 가장 선호되는

〈표 7〉 모형의 적용 결과

〈Table 7〉 The final results of models through their applications

모형	항목	선택안	비고
선형 가중 모형	Erwin ERX		
선형 배정 모형	Bachman's Solution		
가중 차이 모형	Erwin ERX		
타협 해법	Erwin ERX	$p=1, 2, \infty$ 를 사용	
지배 모형	Bachman's, Erwin, Vivid		
최대최대 모형	Vivid Clarity		
사전식 순위법	Erwin ERX	하단 참조 *	
EBA	Bachman's Solution	하단 참조 **	
접속 모형	Erwin ERX	하단 참조 ***	
비접속 모형	Bachman's Solution	하단 참조 ****	
휴리스틱 모형	Erwin ERX	$T_s=0.5, T_g=0.05, T_d=0.5$	

나열 순서는 우선 순위를 나타내고, ()는 허용기준임

*:DLDM, GPDM, GDS, DM, DOC, RLDM, RPDM, SUP, PRC

** :DLDM(6), GPDM(6), GDS(6), DM(5), DOC(6), RLDM(6), RPDM(6), SUP(5), PRC(6)

***:DLDM(6), GPDM(6), GDS(6), DM(5), DOC(5), RLDM(6), RPDM(6), SUP(5), PRC(6)

****:DLDM(6), GPDM(6), GDS(6), DM(5), DOC(6)

대안으로 선정되었고, Bachman's Solution은 4개의 모형에서 선호되고 있다는 것을 보여준다. 특히하게 Vivid도 2개의 모형에서 선호되고 있는 것은 가격 속성이 최고 점수인 10.00 만점을 획득한 결과에 기인한다. S-Designor가 어떠한 모형에서도 선호되고 있지 못한 이유는 Erwin에 의해 지배되고 있기 때문이다.

특히 보상 모형에서는 선형 배정 모형을 제외하고는 Erwin이 최종안으로 채택되어 비교적 일관적인 결과를 보이고 있다. 반면 비보상 모형은 다양한 대안이 최종안으로 선정되고 있는 결과를 보이고 있다. 그 이유는 보상 모형에서는 비교적 대부분의 관련 정보가 모두 활용되고 있기 때문이고, 비보상 모형의 경우는 다양한 선정 규칙과 허용기준이 존재하기 때문이다.

종합적으로 분석결과를 관찰하면 적용 모형에 따라 다양한 소프트웨어 제품이 선정되고 있음을 알 수 있다. 이것은 소프트웨어 제품을 평가·선정함에 있어 하나의 모형만을 사용하는 것이 매우 위험스러울 수

있다는 것을 암시한다. 따라서 사용자는 의사결정 지원을 위한 보다 많은 정보를 얻기 위하여 보상·비보상 모형을 동시에 적용하는 것이 바람직하다. 그 이유는 다른 한 모형이 제공하지 못하는 정보를 얻을 수 있기 때문이다. 이러한 의미에서 두 유형의 모형은 서로 배타적인 관계가 아니라 합리적 평가·선정을 위한 보완적 관계라 할 수 있다.

6. 결 론

지금까지 소프트웨어 제품의 선정을 위한 절차를 2 단계로 구분하여 적용하였다. 가중치는 최종 결과에 상당한 영향을 미치기 때문에 일관적이고 객관적이어야 한다. 가중치의 산정을 위하여 사용된 AHP는 일관성과 객관성의 유지를 위하여 매우 유효한 것으로 활용이 되고 있다.

소프트웨어 제품의 평가·선정의 과정은 복잡하고, 사용자의 욕구가 다양한 바, 유일하고 보편적인 모형을 제시한다는 것이 불가능하다. 이러한 이유로 모형의 성격상 소프트웨어 평가에 알맞은 모형들을 추출하여 보상·비보상 모형으로 구분하여 비교하였다. 상기의 응용에서 알 수 있듯이 선정된 모형에 실제 자료를 투입하여 적용한 결과를 보면 보상 모형과 비보상 모형의 적용에서 많은 차이가 나타남을 알 수 있다. 이러한 차이는 모형이 가지고 있는 특성과 사용자의 선정목적의 상이함에서 연유한다고 할 수 있다.

보상 모형의 장점은 각 제품에 대한 유일한 순위와 결과가 가능하며 일반적인 상황하에서 쉽게 최종안을 탐색할 수 있다는 것이다. 그러나 모형의 편협한 평가 관점으로 인하여 실질적인 정보의 활용도가 감소될 수 있으며 의사결정을 무의미하게 만드는 극단적 요인(outlier)의 존재가 간과될 수 있다. 반면 비보상 모형은 사용자의 독특한 요구사항을 만족시킬 수 있는 특정한 상황에 알맞다. 그러나 최종안의 선정이 불가능한 경우와 순환적 과정을 거쳐야만 최종안에 이르게 되는 불편한 점이 있다. 또한 어느 정도 인위적 요소가 개입되는 허용기준의 부여는 최종적인 결과에 상당한 영향을 미치므로 신중해야 한다.

지금까지의 소프트웨어 평가 및 선정에 관한 단편적인 연구는 평가 환경이나 평가 모형의 장·단점에 대한 분석이 없이 소수의 특정 모형을 적용하기 때문

에 평가 결과에 대한 신뢰성이 저하되고 있다. 기존 연구를 보완하기 위하여 본 연구에서는 소프트웨어 제품의 평가 및 적용에 사용될 수 있는 다양한 모형을 보상/비보상적 관점에서 제시·분류하고, 모형의 장 단점을 비교하였다. 또한 실증적 자료를 사용하여 가중치 및 우선순위를 결정하고, 평가목적별로 모형의 적용 결과를 예시하였다. 결과적으로 본 연구는 가중치 결정과 평가모형을 동시에 적용함으로써 대안 평가가 객관적이고, 대안의 선정이 정확하도록 유도하는 역할을 한다. 따라서 사용자는 모형의 장·단점을 비교하여 평가환경 또는 평가목적에 알맞는 모형을 선택하여 사용한다면 본 연구결과의 적용은 사용자의 정보 요구를 충족시킬 수 있는 모형 선정의 범위를 확대시키고, 모형 적용성을 향상시키는 계기가 될 것이다.

참 고 문 헌

- [1] E. E. Anderson, "Choice Models for the Evaluation and Selection of Software Packages," *Journal of Management Information Systems*, Vol.6, No.4, pp.124-138, Spring 1990.
- [2] G. Visker and A. H. Bree, "Standard Software: A Survey," *Information*, Vol.29, No.10, pp.902-907, October 1987.
- [3] H. C. Lucas, E. J. Walton and M. J. Ginzberg, "Implementing Packaged Software," *MIS Quarterly*, Vol.12, No.4, pp.537-549, December 1988.
- [4] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology," *MIS Quarterly*, Vol.13, No.3, pp. 319-339, September 1989.
- [5] G. L. Sanders, P. Ghandforoush and L. M. Austin, "A Model for the Evaluation of Computer Software Packages," *Computer and Industrial Engineering*, Vol.7, No.4, pp.309-315, 1983.
- [6] P. A. Brown and D. F. Gibson, "A Quantified Model for Site Selection-Application to a Multi-plant Problem," *AIIE Trans.*, Vol.4, pp.1-10, March 1972.
- [7] F. Zahedi, "Data-Base Management System Evaluation and Selection Decision," *Decision Sciences*, Vol.16, No.1, pp.91-116, 1985.
- [8] T. L. Saaty, "How to Make a Decision: The Analytic Hierarchy Process," *European Journal of Operational Research*, Vol.48, No.1, pp.9-26, 1990.
- [9] B. Szajna, "Software Evaluation and Choice: Predictive Validation of the Technology Acceptance Instrument," *MIS Quarterly*, Vol.18, No.3, pp. 319-324, September 1994.
- [10] H-W. Jung and M-S. Yoon, "A Software Product Quality Evaluation and Resource Allocation Model," *The 5th European Conference on Software Quality*, pp.286-294, 1996.
- [11] ISO/IEC 9126 Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for Their Use, ISO, 1991.
- [12] E. E. Anderson, "A Heuristic for Software Evaluation and Selection," *Software-Practice and Experience*, Vol.19, No.8, pp.707-717, August 1989.
- [13] ISO/IEC draft DIS 14598-1: Information Technology-Software Product Evaluation-Part 1: General Overview, ISO, May 1996.
- [14] T. L. Saaty and L. G. Vargas, *The Logic of Priorities*(ISBN:0-89838-071-5), Kluwer-Nijhoff Publishing, London, 1982.
- [15] R. P. Minch and G. L. Sanders, "Computerized Information Systems Supporting Multicriteria Decision Making," *Decision Sciences*, Vol.17, pp. 395-413, 1986.
- [16] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.
- [17] A. Teclé, M. Fogel and L. Duckstein, "Multicriterion Selection of Wastewater Management Alternatives," part of the *Journal of Water Resources Planning and Management*, Vol.114, No.4, July 1988.
- [18] I. B. Hong and D. R. Vogel, "Data and Model Management in a Generalized MCDM-DSS," *Decision Sciences*, Vol.22, pp.1-25, 1991.
- [19] A. Tversky, "Elimination by Aspects: A Theory

of Choice," Psychological Review, Vol.79, No.4, pp.281-299, July 1972.

[20] J. H. Engel, R. D. Blackwell and P. W. Miniard, Consumer Behavior, 6th Ed., the Dryden Press, Chicago, 1990.

[21] I. Ali, W. D. Cook and M. Kress, "On the Minimum Violations Ranking of a Tournament," Management Science, Vol.32, No.6, pp.660-672, June 1986.



정 호 원

- 1979년 고려대학교 산업공학과 (학사)
- 1981년 한국과학기술원 산업공학과(공학석사)
- 1990년 University of Arizona 대학원(경영학박사)
- 1982년~1985년 (주)DACOM

근무

1991년~1995년 한국전산원 근무
 1995년~현재 고려대학교 경영학과 부교수
 관심분야: 소프트웨어 품질평가, 네트워크 성능분석, 내부점법 알고리즘



박 호 인

- 1982년 고려대학교 경영학과 졸업(학사)
- 1986년 New York 주립대학교 경영대학원(경영학석사)
- 1997년 고려대학교 대학원 경영학과(경영학박사)
- 1988년~1990년 고려대학교 기업경영연구소 연구원

1990년~현재 부천전문대학 사무자동화과 부교수
 관심분야: 소프트웨어 품질평가, 중소기업 정보화