

소프트웨어 제품 품질 평가를 위한 가중치 측정시스템

구자경*, 김길조*, 안유환*, 김진수**

*한국전자통신연구원 컴퓨터소프트웨어기술연구소 실시간컴퓨팅연구부

**건양대학교 정보전자통신공학부

A Weight Measurement System for Software Product Quality Evaluation

Ja-Kyong Koo*, Gil-Jo Kim*, Yu-Whoan Ahn*, Jin-Soo Kim**

*Dept. of Real-Time Computing, ETRI-Computer & Software Technology Laboratory

**Dept. of Information, Electronics & Communications Engineering, Konyang University

요 약

소프트웨어 제품 품질 측정이란 다양한 소프트웨어의 특성들을 통하여 소프트웨어 제품의 품질을 정확하게 측정하는 것이다. 이를 위해서는 소프트웨어의 종류 및 사용환경에 따라 품질 특성들에 대한 다양한 가중치를 부여하여 평가할 필요가 있다. 본 논문에서는 AHP를 사용하여 품질 특성들에 대한 상대 중요도를 제공하여 특성들에 대한 가중치를 반환하는 가중치 측정시스템을 개발하였다. 본 시스템은 ETRI 소프트웨어 품질보증 연구팀에서 개발하고 있는 소프트웨어 제품 품질 측정도구와 연계되어 사용될 수 있으며 다른 분야에서도 독립적으로 활용될 수 있다.

1. 서 론

소프트웨어 제품 품질 측정이란 다양한 소프트웨어의 특성들을 통하여 소프트웨어 제품의 품질을 정확하게 측정하는 것이다. 이를 위해서는 소프트웨어의 종류 및 사용환경에 따라 품질 특성들에 대한 다양한 가중치를 부여하여 평가할 필요가 있다. 또한 사용자나 평가자에 따라 평가를 원하는 특성들을 대상으로만 평가를 하기도 한다. 본 논문에서는 이러한 다양한 특성들에 대한 평가 가중치를 알기 위해서 각 평가 항목들의 상대 중요도를 제공하여 평가에 필요한 항목들의 가중치를 구할 수 있는 시스템을 개발하였다. 본 시스템은 [1]에 기술된 Analytical Hierarchy Process(AHP)를 사용하여 품질 특성들에 대한 상대 중요도를 입력받아 특성들에 대한 가중치를 반환하는 가중치 측정시스템이다. 본 시스템은 ETRI 소프트웨어 품질보증 연구팀에서 개발하고 있는 소프트웨어 제품 품질 측정도구와 연계되어 개발되었으며 다른 분야에서도 독립적으로 활용될 수 있다. 2장에서는 AHP의 단계를 소개하고 3장에서 AHP를 이용한 가중치 측정시스템을 설명한 후 4장에서 결론을 맺는다.

2. Analytical Hierarchy Process

AHP는 어떠한 결정을 하는데 도움을 주기 위하여 필요로 하는 여러 항목들의 공헌도를 단계적으로 측정하기 위하여 개발된 방법이다[2]. 본 시스템은 이러한 AHP를 소프트웨어 제품의 품질 평가를 위하여 요구되는 각 품질 특성 및 부특성들의 가중치를 계산하는데 사용하였다. AHP를 이용하여 가중치를 계산하기 위해서는 다음과 같은 네단계가 필요하다.

2.1 측정 대상 설정

먼저 측정을 하고자 하는 항목을 설정하고 그 항목의 수에 따라 $n \times n$ 매트릭스를 구성한다. 예를들어, ISO/IEC 9126에서는 6가지 항목의 소프트웨어 품질 특성(외부특성)이 있고 이를 좀더 세분화한 21개의 부특성들이 있는데 이 가운데 기능성의 부특성으로는 적합성, 정확성, 상호운용성, 보안성, 준수성의 다섯가지가 있다[3]. 이러한 다섯가지의 부특성들에 대한 가중치를 계산하고자 한다면 5×5 매트릭스가 필요하게 된다.

2.2 상대 비교 수행

다음으로는 일정한 기준에 따라 모든 항목들의 상대 비교를 수행한다. 여기에서 사용되는 기본적인 값은 동등하다고 생각되면 1 값을 부여하고, 다른 하나보다 경험적으로 약간 우세하다고 생각되면 3 값을 부여한다. 다른 하나보다 경험적으로 많이 우세하다고 생각되면 5 값을 부여하고, 다른 하나보다 많이 우세하고 그 우세가 실제로 증명이 된다면 7 값을 부여한다 9 값은 다른 하나보다 아주 많이 확실히 우세할 경우 부여하게 된다 두 값의 중간 값을인 2, 4, 6, 8도 사용될 수 있다.

표 1. 각 항목들에 대한 상대 비교

	적합성	정확성	상호운용성	보안성	준수성
적합성	1	3	1	3	5
정확성	1/3	1	1/3	5	5
상호운용성	1	3	1	3	3
보안성	1/3	1/5	1/3	1	1
준수성	1/5	1/5	1/3	1	1

대각선을 기준으로 매트릭스의 우측 상단부분에 상대 값을 주게 되면 좌측 하단부분은 그 값들의 역을 취하여 구할 수 있고 대각선은 모두 1 값을 갖는다

2.3 가중치의 측정

매트릭스의 각 항목에 대한 가중치를 구하기 위해서는 정규화된 열에 대한 평균을 사용한다. [1]에서 제공하는 방법으로 먼저 매트릭스에서 각 열의 합을 계산한다. 다음으로 매트릭스에서 각 원소를 원소가 속해 있는 열의 합으로 나누고 각 행의 합을 계산한다.

표 2 매트릭스의 각 원소에 대한 계산 값

	적합성	정확성	상호운용성	보안성	준수성	합계
적합성	0.35	0.41	0.33	0.23	0.33	1.65
정확성	0.12	0.14	0.11	0.38	0.33	1.08
상호운용성	0.35	0.41	0.33	0.23	0.2	1.52
보안성	0.12	0.03	0.11	0.08	0.07	0.41
준수성	0.07	0.03	0.11	0.08	0.07	0.36

다음으로 각 행의 합을 항목의 수로 나누어서 행들의 합을 정규화한다. 이 계산의 결과가 매트릭스의 각 항목에 대한 가중치가 된다.

$$\frac{1}{5} \cdot \begin{pmatrix} 1.65 \\ 1.08 \\ 1.52 \\ 0.41 \\ 0.36 \end{pmatrix} = \begin{pmatrix} 0.33 \\ 0.22 \\ 0.30 \\ 0.08 \\ 0.07 \end{pmatrix}$$

위의 결과를 보면 적합성은 전체 기능성의 33 %의 가중치를 가지게 되고 정확성은 22 %, 상호운용성은 30 %, 보안성은 8 %, 준수성은 7 %의 가중치를 가진다고 할 수 있다.

2.4 일관성 검사

일반적으로 모든 항목들에 대한 상대 가치를 정확하게 결정할 수 있다면 가중치는 완전하게 일관성이 있을 것이다. 예를들어, 우리가 적합성이 정확성보다 더 가치가 있고, 정확성은 상호운용성보다 다소 더 가치가 있고, 상호운용성은 적합성보다 약간 가치가 있다고 결정한다면 이 세항목은 서로 일관성이 없게 되고 결과로 얻어진 가중치들의 정확성은 감소하게 된다 따라서 계산된 가중치들을 이용하여 각 항목들의 일관성을 검사할 필요가 있다 이러한 일관성 검사를 위해서는 일관성 인덱스와 일관성 비율을 계산해야하고 이 계산의 결과를 통하여 판단의 오류를 측정하도록 해준다[4]

일관성 인덱스 CI(Consistency Index)는 $(\lambda_{max}-n)/(n-1)$ 로써 계산되며 λ_{max} 의 값이 항목의 수와 가까우면 가까울 수록 판단 오류는 적어지고 따라서 결과는 좀더 일관성이 있게 된다. λ_{max} 를 측정하기 위해서는 먼저 매트릭스를 각 항목의 가중치로 곱한다.

$$\begin{pmatrix} 1 & 3 & 1 & 3 & 5 \\ 1/3 & 1 & 1/3 & 5 & 5 \\ 1 & 3 & 1 & 3 & 3 \\ 1/3 & 1/5 & 1/3 & 1 & 1 \\ 1/5 & 1/5 & 1/3 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0.33 \\ 0.22 \\ 0.30 \\ 0.08 \\ 0.07 \end{pmatrix} = \begin{pmatrix} 1.88 \\ 1.18 \\ 1.74 \\ 0.40 \\ 0.36 \end{pmatrix}$$

다음으로 각 항목의 가중치로 결과 벡터의 각 원소를 나눈다.

$$\begin{pmatrix} 1.88 / 0.33 \\ 1.18 / 0.22 \\ 1.74 / 0.30 \\ 0.40 / 0.08 \\ 0.36 / 0.07 \end{pmatrix} = \begin{pmatrix} 5.70 \\ 5.36 \\ 5.80 \\ 5.00 \\ 5.14 \end{pmatrix}$$

λ_{max} 를 계산하기 위해서는 결과 벡터에서 원소들의 평균을 구해야 한다.

$$\lambda_{max} = \frac{5.70+5.36+5.80+5.00+5.14}{5} = 5.4$$

이 λ_{max} 를 이용하여 일관성 인덱스를 계산하면 다음과 같다.

$$CI = \frac{\lambda_{max}-n}{n-1} = \frac{5.4-5}{5-1} = 0.1$$

이러한 일관성 인덱스가 받아들여질 수 있다는 것을 확인하기 위해서는 다음과 같이 일관성 비율을 계산해야 한다

임의로 생성된 반대쪽 매트릭스의 일관성 인덱스를 임의 인덱스(Random Index)라고 한다. 같은 차수의 매트릭스를 위한 RI에서 CI의 비율을 일관성 비율(Consistency Ratio)이라 하고 상대 비교의 정확성을 정의한다. 5 x 5 매트릭스를 위한 RI는 1.12가 된다. 따라서 CR은 다음과 같다.

$$CR = \frac{CI}{RI} = \frac{0.1}{1.12} = 0.09$$

일반적으로 0.10 이하의 일관성 비율이 되어야만 입력된 상대 가치들이 일관성이 있다고 간주된다

3. 가중치 측정시스템

2장에서 설명한 AHP를 이용하여 소프트웨어 품질 특성들의 가중치를 구하는 단계를 적용하여 가중치를 측정하는 시스템을 구현하였다. 본 시스템은 품질평가 도구와 연계되어 사용되며 평가도구에서 제공하는 항목 및 항목들의 정보를 기반으로 사용자에게 각 항목들의 상대 값을 입력받아 가중치를 계산하게 되고 그 결과를 다시 품질평가 도구로 돌려주게 된다. 먼저 본 시스템의 구조를 보면 다음 그림 1과 같다.

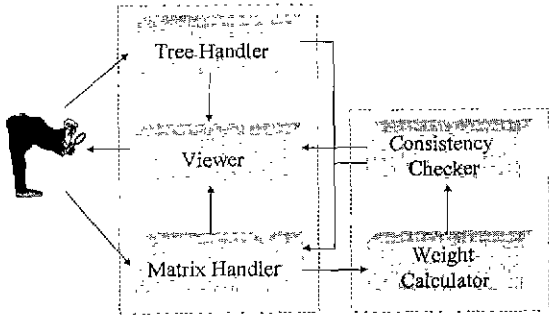


그림 1 측정 시스템의 구조

(1) Tree Handler

사용자가 측정하고자 하는 항목들의 계층 구조를 트리 형태로 보여준다. 가중치를 측정하고자 하는 노드들의 한 단계 상위 노드를 클릭하면 해당 항목들의 리스트는 뷰어를 통하여 화면에 보여지게 되고 Matrix Handler에게 각 항목의 노드명을 제공한다.

(2) Viewer

사용자에게 트리 구조, 리스트, 매트릭스 및 계산된 가중치를 보여준다.

(3) Matrix Handler

사용자에게 각 항목의 차수에 맞는 매트릭스를 제공하고 사용자로부터 매 매트릭스의 우측 상단부분에 각 항목들의 상대 값을 직접 입력할 수 있도록 하고 입력을 완료한 후 Calculate 버튼을 클릭하여 가중치를 계산하도록 한다.

(4) Weight Calculator

AHP의 가중치 계산 방법에 따라 각 항목들의 가중치를 계산하게 된다. 계산된 각 항목의 가중치는 뷰어를 통하여 사용자에게 보여지게 된다.

(5) Consistency Checker

계산된 각 항목의 가중치들이 일관성이 있는지를 검사한다. 만약 계산된 각 항목들간의 일관성이 없으면 재입력을 요구하게 된다.

그림 2는 가중치 측정시스템의 실행화면으로 좌측부분은 각 항목들이 트리 형태로 보여지게 되고, 우측 부분의 상단

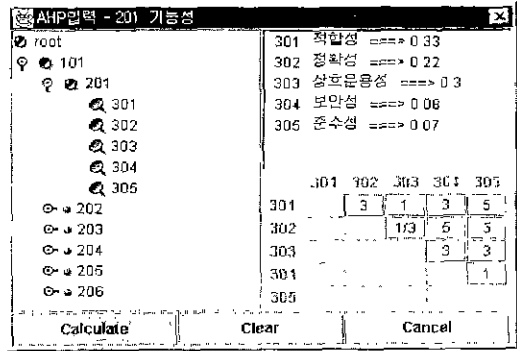


그림 2. 가중치 측정시스템의 실행화면

에는 각 항목들의 정보와 계산된 가중치가 보여진다. 우측 하단에는 선택된 항목들의 가중치를 계산하기 위한 매트릭스가 제공되고 사용자는 이 매트릭스의 대각선을 기준으로 우측 상단부분에 직접 상대 값들을 입력할 수 있다. 입력을 완료하고 Calculate 버튼을 누르면 가중치를 구하게 되고, Clear 버튼을 누르면 현재 입력한 데이터가 지워진다. 또한 Cancel 버튼을 누르면 모든 과정을 취소하게 된다.

4. 결론

본 논문에서는 소프트웨어 제품의 품질을 측정하는데 있어서 소프트웨어의 종류 및 사용환경에 따라 품질 특성들을 선택하여 선택된 특성들을 대상으로 가중치를 부여하기 위하여 AHP를 이용한 가중치 측정시스템을 제안하였다. 본 시스템은 사용자나 평가자가 평가를 원하는 특성에 대한 정보 및 상대 중요도를 제공하여 품질 평가에 필요한 항목들의 가중치를 구할 수 있도록 하였다. 본 시스템은 ETRI 소프트웨어 품질보증 연구팀에서 개발하고 있는 소프트웨어 제품 품질 측정도구와 연계되어 개발되었고 다른 분야에서도 독립적으로 활용될 수 있을 것이다.

참고 문헌

- [1] T. L. Saaty, The Analytical Hierarchy Process, McGraw-Hill, 1980
- [2] Joachim Karlsson and Kevin Ryan, A Cost-Value Approach for Prioritizing Requirements, IEEE Software, 1997
- [3] ISO/IEC 9126-1, Information Technology - Software Product Quality - Part 1 Quality model
- [4] 박호인, 정호원, "소프트웨어 제품을 위한 평가선정 모형의 조사 및 적용성에 관한 연구", 경보처리논문지, vol.4. No 7, July, 1997
- [5] Andre Koscianski and Joao Candido Bracarense Costa, Combining Analytical Hierarchical Analysis with ISO/IEC 9126 for a Complete Quality Evaluation Framework, 4th IEEE International Symposium and Forum on Software Engineering Standards, 1999