

소프트웨어의 계층적 분류에 기반한 품질 인증

박동철^{0*} 김순용^{**} 이상덕^{**} 오재원^{*} 이종원^{*} 우치수^{*}

^{*}서울대학교 컴퓨터공학부

^{**}한국정보통신기술협회

^{*}(tongss⁰, jwoh, jwlee, wuchisu)@selab.snu.ac.kr

^{**}(syk, sdlee)@tta.or.kr

Software certification based on hierarchical classification of software

Dongchul Park^{0*} Soon-Yong Kim^{**} Sang-Duck Lee^{**} Jaewon Oh^{*} Jongwon Lee^{*} Chisu Wu^{*}

^{*}School of Computer Science & Engineering, Seoul National University

^{**}Telecommunications Technology Association

요 약

오늘날 소프트웨어는 개발 방법론을 이용하여 양적으로나 질적으로 많은 발전이 이루어졌다. 이와 더불어 소프트웨어에 대한 소비자들의 요구와 지식 수준 또한 같이 높아짐으로써 소프트웨어 인증에 대한 관심과 인식도 같이 높아지고 있는 추세이다. 본 논문에서는 소프트웨어 품질 인증을 위한 메타 모델의 필요성을 알아보고 기존 연구[1]되었던 메타 모델의 개념 수준을 더욱 발전시켰다. 그리고 간단하지만 실질적인 예제를 통해서 메타 모델과 인증 모델, 인증 프로그램과의 관계를 명확하게 정립하고자 한다.

1. 서 론

정보산업사회로 탈바꿈하면서 소프트웨어 기술의 산업적 중요성이 증대하고 국제 무역 거래 시 각국의 소프트웨어 인증 요구가 증가함에 따라 국내에서도 소프트웨어 제품의 결함을 최소화하여 품질 보증을 할 수 있는 기술 확보가 시급하다. 미국, 유럽의 경우 소프트웨어 개발 기업의 품질 경영 시스템을 평가하는 CMM, TL9000 인증을 요구하고 있으며 일본은 정량적 품질 관리 기술을 적용하여 소프트웨어 제품의 결함을 최소화한 품질 보증을 요구하고 있다[2].

이와 관련하여 현재 소프트웨어 인증 현황을 살펴보면 소프트웨어 인증은 크게 개발 프로세스 인증과 제품 품질 인증으로 나뉘어진다. 먼저 프로세스 인증 방식은 소프트웨어 개발 프로세스를 인증하는 방식으로 의료용 시스템이나 원자력 발전 시스템과 같이 높은 안전성을 요구하는 분야에서 많이 사용하는 인증 방식이며 전 세계적으로 CMM, SPICE, TL9000등이 있다. 이에 반해 완제품 인증 방식은 소프트웨어 완제품 자체를 인증하는 것으로 아직까지 국제적인 인증기준은 없으나 ISO 9126[3]을 기반으로 한 품질 인증 사례들이 이런 접근 방법에 속한다.

일반 사용자들이 접하는 대부분의 소프트웨어는 체계적인 개발 프로세스를 갖추고 있지 않은 소규모 기업들에서 개발된 것으로 프로세스 인증 방식을 채택하기에는 문제가 있다[1]. 따라서 본 논문에서는 소프트웨어 인증을 위해서 제품 인증 방식을 채택하고자 한다.

2. 연구의 필요성

소프트웨어는 유형별로 서로 다른 인증 방법(certification methodology)을 요구한다[4]. 또한 사용 환경이나 소프트웨어의 특성에 따라 품질을 평가하기 위한 메트릭의 선택이 달라질 수도 있다[5][6][7]. 예를 들어, 네트워크 프로그램의 신뢰성을 데이터베이스 프로그램의 신뢰성과 같은 메트릭으로 평가할 수는 없다. 따라서 해당 소프트웨어를 좀더 정확하게 인증하기 위해서는 유형별로 소프트웨어를 체계적으로 분류하고 사용 환경이나 소프트

웨어의 특성을 고려할 필요가 있다.

오늘날 소프트웨어는 개발 방법론을 이용함으로써 새로운 유형의 소프트웨어를 포함하여 소프트웨어가 급진적으로 증가하고 있다. 그리고 소프트웨어가 단순히 컴퓨터 영역뿐만 아니라 여러 일상적인 영역으로까지 사용이 확대되고 있으며 또한 COTS(Commercial Off The Shelf) 소프트웨어의 급격한 증가로 인하여 새로운 인증 방법과 인증 프로그램을 보다 적은 비용으로 신속하게 개발할 필요성을 느끼게 되었다. 이에 본 논문에서는 품질 인증을 위한 메타 모델을 제안하게 되었다.

메타 모델은 위에서 언급한 소프트웨어의 시장 동향에 맞추어 새로운 소프트웨어 품질 인증 방법과 인증 프로그램을 보다 적은 비용으로 신속하고 편리하게 개발할 수 있도록 고안된 모델이다. 본 논문에서는 지금까지 개념 확립 수준이었던 메타 모델을 적용하여 인증 모델, 인증 프로그램의 생성까지 발전시켰다. 그리고 실제 적용 예제를 보임으로써 메타 모델의 적용 방안을 생각해보고 메타 모델과 인증 모델, 인증 프로그램과의 관계를 명확하게 정립하고자 한다.

3. 소프트웨어 품질 인증을 위한 메타 모델

인증 기관은 메타 모델을 이용하여 인증 모델과 인증 프로그램을 생성하고 그 인증 프로그램을 이용하여 해당 소프트웨어를 인증할 수 있다. 이 장에서는 그러한 메타 모델의 정의와 그 구성 요소들에 대해서 간단하게 언급하고, 메타 모델 적용 방법에 주안점을 두겠다.

3.1 품질 인증 메타 모델의 정의

인증 메타 모델이란 인증 기관이 인증 방법, 인증 대상, 인증 범위 등을 결정하기 위해서 참조하는 모델로서 인증 프로그램을 체계적으로 생성하기 위한 규칙(Rule)을 포함하는 개념이다. 즉, 인증 기관은 인증 서비스 전에 인증 모델을 정립하여 이를 기반으로 인증과 관련된 전반적인 요소를 결정하게 되고 인증 프로세스 전체에 걸쳐 만들어진 인증 모델을 사용하게 된다.

3.1.1 메타 모델의 세 가지 요소

인증 메타 모델은 품질요소, 소프트웨어 도메인, 소프트웨어 사용환경으로 이루어져 있으며 이 세 가지 요소들이 소프트웨어 품질 인증의 세 가지 요소가 된다.

● 소프트웨어 도메인

소프트웨어 도메인이란 소프트웨어 분류체계를 말한다. 소프트웨어 품질 측정 및 인증 과정은 해당 소프트웨어의 종류에 따라서 달라질 수 있으므로 소프트웨어 분류는 품질 인증에서 중요한 부분을 차지한다. 그러나 소프트웨어 분류에 대해 명확하게 정립된 하나의 표준이 존재하지는 않는다. ISO 12182[8]를 비롯해서 소프트웨어 분류와 관련된 여러 가지 문서들을 살펴보면 일반적으로 분류체계는 계층성을 가지고 있음을 알 수 있다. 여기에서 계층성이란 하위 분류가 상위 분류의 부분집합으로 이루어지는 IS-A 관계를 말하는 것으로 소프트웨어 도메인의 중요한 성질이 된다. 이렇듯 소프트웨어 도메인은 계층적 구조를 가지고 있어야 하며 또한 향후 새로운 소프트웨어를 대비한 확장성을 지니고 있어야 한다.

● 품질 요소

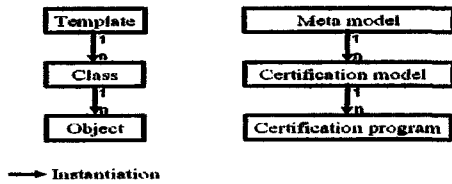
소프트웨어 품질은 단일하게 표현되는 것이 아니다. 일반적으로 세부 요소를 평가하여 그것들의 종합으로 나타내는 것이 일반적이다. 따라서 품질 요소는 품질 평가의 기본 단위로써 품질 평가는 편의성, 가용성 등 품질 요소를 단위로 먼저 이루어져야 한다[3].

● 소프트웨어 사용환경

여기에서 말하는 환경이란 소프트웨어가 최종적으로 사용되는 실제 환경을 말한다. 즉, 사용환경에 따라 매트릭의 선택이 달라질 수 있으며 또한 매트릭의 하한값이 달라질 수도 있으므로 인증에는 소프트웨어 사용환경이 고려되어야 한다[5].

3.1.2 메타 모델, 인증 모델과 인증 프로그램의 관계

템플릿이 객체를 생성하는 클래스의 상위 개념인 것처럼, 메타 모델은 특정 인증 프로그램 생성 이전에 구축되는 인증 모델을 도출하기 위한 상위 개념인 틀이라고 할 수 있다. 따라서 인증 기관은 해당 소프트웨어 인증 환경이 변경되더라도 유연한 대처가 가능하다 (<그림 1> 참조).



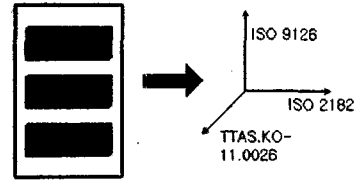
<그림 1> 메타 모델, 인증 모델, 인증 프로그램 상관도

3.2 품질 인증 메타 모델의 적용

메타 모델의 적용은 크게 인증 모델의 생성과 인증 프로그램의 생성 두 단계로 나눌 수 있다. 여기에서는 인증 메타 모델을 이용하여 인증 모델과 인증 프로그램을 생성하는 예제를 통해서 메타 모델의 적용 방법을 알아본다.

3.2.1 인증 모델의 생성

C++의 템플릿을 이용하여 여러 가지 클래스가 생성될 수 있는 것처럼 메타 모델의 세 요소를 무엇으로 결정하느냐에 따라서 여러 가지 인증 모델이 생성될 수 있다. 여기에서는 소프트웨어 도메인으로 한국정보통신기술협회(TTA)의 소프트웨어 재사용 분류체계표준(TTAS.KO-11.0026)[9]을 사용하고, 품질요소로는 ISO/IEC9126[3], 소프트웨어 사용환경으로 ISO/IEC 12182[8]를 사용한 인증 모델을 예제로 사용하기로 한다(<그림 2> 참조).



③ 메타 모델 ④ 인증 모델
<그림 2> 메타 모델을 적용한 인증 모델

단, 소프트웨어 사용환경에 적용한 ISO/IEC 12182는 소프트웨어 분류와 관련된 문서로 여기에서는 소프트웨어 사용환경에 맞게 적용시켰다.

3.2.2 인증 프로그램 생성

인증 모델이 만들어지면 그 인증 모델을 이용하여 실질적으로 인증에 사용되는 인증 프로그램을 생성할 수 있다. 여기에서 품질 요소별 매트릭을 선출하고 그 선출된 매트릭의 하한값을 포함하는 체크 리스트를 작성하게 된다.

3.2.2.1 인증 프로그램 생성 규칙

인증 프로그램의 생성 규칙으로는 다음과 같은 세 가지 방법을 생각해 볼 수 있다.

● Ad hoc approach

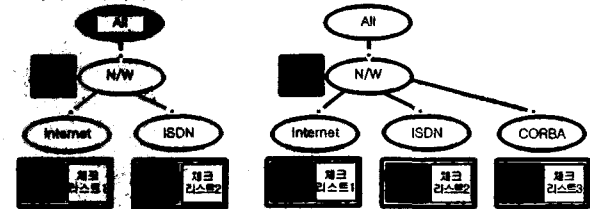
소프트웨어가 입력될 때마다 해당 소프트웨어를 위한 체크 리스트를 작성하는 방식이다. 같은 그룹에 속한 소프트웨어라고 하더라도 매번 리스트를 작성해야 하므로 비효율적이다.

● Group-based approach

각 소프트웨어 그룹에 대해 미리 체크리스트를 마련해서 인증을 위해 소프트웨어가 입력되면 해당 그룹의 체크리스트를 참조하는 방식이다. 이 방식은 하나의 상위 그룹에 속하는 하위 그룹들이 공통적인 부분을 많이 공유하고 있음에도 불필요하게 체크리스트를 각기 생성하여 보유하는 단점을 지니고 있다.

● Group-based approach with specialization

3.1.1에서 언급한 바와 같이 소프트웨어 분류 체계가 가져야 할 특성중의 하나는 계층성이다. 이 기법은 계층성을 잘 반영하여 체크리스트를 체계적으로 유지하고 관리하는 방식으로 기본적으로는 Group-based approach에 기반을 두고 있지만 다음과 같은 차이점이 있다. 즉, 특정 그룹에 대한 체크리스트 작성은 해당 상위 그룹의 체크리스트와 무관한 것이 아니라 상위 그룹의 체크리스트에 기반해서 자신만의 필요한 체크리스트를 추가하여 작성할 수 있다. <그림 3>의 ⑤에서 보듯 네트워크(N/W)의 하위 그룹의 체크리스트는 해당 상위 그룹의 공통 체크리스트(체크리스트0)에 각 하위 그룹에서 필요한 리스트만 추가함으로써 체크리스트를 작성할 수 있다. 또한 이 방법은 새로운 소프트웨어 그룹이 추가되더라도 어려움 없이 확장을 할 수 있는 장점이 있다. <그림 3>의 ⑥에서 CORBA라는 하위 그룹이 추가되더라도 기존의 체계를 유지하면서 어려움 없이 확장 가능한 것을 볼 수 있다.



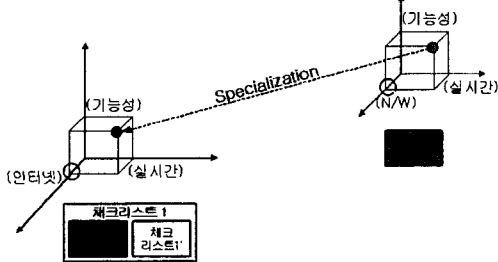
⑤ 기존 분류 ⑥ 새로운 그룹 추가

<그림 3> Group-based approach with specialization

따라서 메타 모델에서는 세 번째 방식을 인증 프로그램 생성 규칙으로 사용하도록 한다.

3.2.2.2 Specialization

앞서 소프트웨어 도메인은 계층적으로 세분화가 되고 소프트웨어는 품질 요소별로 평가를 한다고 하였다. 이런 가정 하에서 Specialization란 <그림 4>에서 보듯 계층적으로 체크리스트를 선출하고 종합하는 것을 말한다. 즉, 체크리스트1은 공통 체크리스트인 상위 그룹의 체크리스트0에 자기만의 고유한 체크리스트1'을 추가하여 만들어 질 수 있다.



<그림 4> Specialization

이렇듯 체크리스트에서도 상속의 개념이 그대로 적용이 되는 데 하위 리스트는 공통된 성질을 가진 상위 메트릭 뿐만 아니라 자기만의 고유한 메트릭을 가질 수가 있다.

그리고 <표 1>은 메트릭과 하한값 그리고 속성으로 이루어진 체크리스트의 구조를 보여준다.

<표 1> 체크리스트 구조 (그림 4의 체크리스트 0)

Metric	Value(하한값)	Property
M1	A	Necessary
M2	B	Optional
M3	C	Necessary

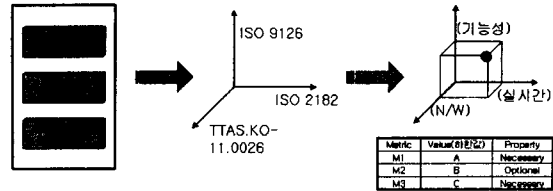
여기에서 메트릭은 품질 요소를 측정하기 위한 방법을 말하며 그 하한값을 통해서 품질 요소의 평가가 이루어진다. 그리고 속성(Property)이란 해당 품질 요소 측정을 위해서 그 메트릭이 반드시(Necessary) 필요하지 아니면 선택적(Optional)인지를 나타내는 요소로서 하한값과 마찬가지로 도메인의 세분화 과정에서 값의 변경이 이루어 질 수도 있다. 해당 품질 요소를 측정하는 메트릭은 여러 가지가 있으나 리스트에 모두 추가되는 것은 아니며 측정 불가능하거나 불필요한 메트릭은 제외된다.

체크리스트의 Specialization 과정에서 앞서 말했듯이 메트릭의 하한값과 속성의 값이 변경될 수가 있는데 이는 객체 지향 언어에서 상속이 가지는 특성중의 하나인 오버라이딩의 개념과 유사하다.

<표 2> 체크리스트 구조 (그림 4의 체크리스트 1)

Metric	Value(하한값)	Property
M5	E	Optional
M6	F	Optional

<표 1>과 <표 2>에서 보듯 하위 체크리스트에 새로운 M4, M5 두 개의 메트릭이 추가되었으며 상위 체크리스트(<표 1>)의 M1의 하한값과 M2의 속성이 각각 하위 체크리스트에서 변경이 되었음을 알 수 있다. 이는 상위 분류에서 하위 분류로 세분화되면서 메트릭의 중요도가 달라질 수 있는 가능성을 반영한 것이다.



<그림 4> 메타 모델, 인증 모델, 인증 프로그램과의 관계

<그림 4>는 지금까지 설명했던 메타 모델의 전반적인 관계를 한눈에 보여준다. 이와 같이 메타 모델은 실제 인증에 사용되는 모델이 아닌 인증 모델을 생성하기 위한 하나의 틀이 되며, 그 틀에 실질적인 요소를 적용하여 하나의 인증 모델이 만들어지게 된다. 또한 그 인증 모델을 이용하여 인증 프로그램이 생성되고 그 인증 프로그램을 이용하여 실제 해당 소프트웨어 품질 인증을 수행하게 되는 것이다.

4. 결 론

지금까지 메타 모델의 예제를 통해서 메타 모델과 인증 모델, 인증 프로그램과의 관계를 알아보았다.

소프트웨어는 유형별로 서로 다른 인증 방법을 요구하며 사용환경 또는 소프트웨어 특성에 따라 품질 평가 방법이 달라질 수 있다. 그리고 다양한 새로운 유형의 소프트웨어 출시와 소프트웨어의 사용 영역의 확장 및 COTS 소프트웨어의 급진적 증가로 새로운 인증 방법을 보다 적은 비용으로 신속하게 개발할 필요성을 느끼게 되었다.

이에 본 논문에서는 품질 인증을 위한 메타 모델을 제안하였다. 메타 모델이란 인증 기관이 인증 모델, 인증 프로그램을 결정하기 위해서 참조하는 모델로서 인증 프로그램을 체계적으로 생성하기 위한 규칙을 포함하는 개념이다. 이러한 메타 모델의 새 요소에 적당한 기준을 적용함으로써 인증 모델과 인증 프로그램을 생성할 수가 있다. 그러나 인증 프로그램의 생성 과정에서 품질 요소에 적합한 메트릭을 선택하거나 그 메트릭의 하한값을 결정하는 일은 품질 인증에 직접적인 영향을 주는 민감한 사항인 만큼 결코 간단한 일이 아니다. 이러한 부분들은 실제 수많은 인증 작업을 통해 얻어질 수 있는 사항들이다.

참고 문헌

- [1] 이종원 등, "소프트웨어 품질 인증을 위한 메타 모델의 적용," 2002 정보과학회 봄 학술발표논문집, 제 29권 1호, 2002.
- [2] 산업자원부, "http://www.ats.go.kr/contents/g_2_4.asp".
- [3] ISO/IEC, "DTR 9126-2: Software Engineering - Product Quality Part 2 - External Metrics," ISO/IEC JTC1/SC7 N2419, 2001.
- [4] Voas, J. M., "Limited software warranties," 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems, 2000, pp. 56-61.
- [5] Bevan, N., "Measuring usability as quality of use," Software Quality Journal, Vol. 4, 1995, pp. 115-150.
- [6] Bevan, N., "Usability is Quality of Use," 6th International Conference on Human Computer Interaction, 1995.
- [7] 최은만 등, "소프트웨어 제품 메트릭과 데이터분석 기술에 관한 연구," 소프트웨어공학회지, 제 14권 4호, 한국정보과학회, 2001, pp. 62-73.
- [8] ISO/IEC, "Information technology - Categorization of software," ISO/IEC TR 12182:1998(E), 1998.
- [9] 정보통신단체표준, "소프트웨어 재사용 분류체계표준," TTAS.KO-11.0026, 한국정보통신기술협회, 2000.