

# 산업용 소프트웨어 국제표준 적합성 인증 시스템의 설계 및 구현

양 해 슨<sup>†</sup> · 최 민 용<sup>††</sup> · 박 인 수<sup>†††</sup>

## 요 약

최근 산업분야 제품의 양적, 질적 확대와 더불어 이와 관련된 응용 기술의 사용이 증가하고 있다. 그 중의 하나가 산업분야에서 사용되는 소프트웨어라 할 수 있는데, 이러한 소프트웨어는 현재 산업용 장비를 활용하는데 있어 가장 중요시되는 부분이라 할 수 있다. 따라서 산업용 장비를 이용하여 해당 작업을 수행하는데 있어 두뇌역할을 담당하는 소프트웨어는 그것의 품질에 따라 해당장비의 성능을 나타낸다 할 수 있다. 이러한 시점에서 산업용 장비에 내장되는 소프트웨어의 품질에 대한 평가와 측정 그리고 이를 통한 품질향상의 의식들이 고취되고 있으며 이러한 요구에 따라 관련된 노력들이 국·내외 적으로 이루어지고 있다. 이에 본 논문에서는 소프트웨어 품질을 측정, 평가하기 위하여 소프트웨어 일반적 품질 요구사항을 위한 국제표준인 ISO/IEC 12119, 소프트웨어 품질특성 및 부특성 평가를 위한 국제표준인 ISO/IEC 9126을 바탕으로 평가 메트릭을 구축하였고, 이의 수행을 위하여 평가모듈의 구성을 위한 국제표준인 ISO/IEC 14598-6을 바탕으로 산업용 소프트웨어의 정량적인 품질을 측정, 평가하여 품질측정의 결과를 토대로 품질을 인증하는 산업용 소프트웨어 국제표준 적합성 인증 시스템을 설계 및 구현하였다.

## Design and Implementation of Compatible Certification System of International Standard based Industrial Software

Hae-Sool Yang<sup>†</sup> · Min-Yong Choi<sup>††</sup> · In-Soo Park<sup>†††</sup>

## ABSTRACT

In the latest, it's increasing an applied technology to be related with growth of industry circles. It's one of them a software to be used in industry circles, and It's most important part to apply an industrial equipment, so software to take charge of major part is indicative performance of equipment. At this time, it's inspired an evaluation and measurement of software quality to have within industrial equipment, and it's forming the research and development by the inside and outside of the country. For this, it's constructed a valuation metric to be based on ISO/IEC 12119, the International Standard for general the terms desired of quality of software and ISO/IEC 9126-2, the International Standard of the terms valuation of quality for evaluation and measure, and for this accomplishment, It has been designed and developed industrial software international standard compatible approval system which approve a quality based on quality test result of industrial software using the ISO/IEC 14598-6 that international standard for organization of evaluation module.

**키워드 :** 소프트웨어 품질(Software Quality), 국제표준(ISO : International Standardization Organization), 산업용 소프트웨어 국제표준 적합성 인증 시스템(Industrial Software International Standard Compatible Approval System), 측정(Measurement), 평가(Valuation)

### 1. 서 론

최근 산업분야 제품의 양적, 질적 확대와 더불어 이와 관련된 응용기술의 사용이 증가하고 있다. 이러한 응용기술의 사용은 제품을 사용하는 사용자의 편의를 도모하고 기존의 제품에 비하여 고성능의 작업수행능력을 보유하기 위한 것으로 그 중의 하나가 제품을 운용, 제어하는 소프트웨어라 할 수 있는데, 이러한 소프트웨어는 현재 산업용 장비를 활

용하는데 있어 가장 중요시되는 부분이라 할 수 있다. 따라서 산업용 장비를 이용하여 해당 작업을 수행하는데 있어 두뇌역할을 담당하는 소프트웨어는 그것의 품질에 따라 해당장비의 성능을 나타내며 그의 성능에 따라 우리의 생활에 미치는 영향이 크게 좌우된다고 할 수 있다[2].

실제 2000년 밀레니엄 버그와 같은 문제는 전 세계적으로 큰 사회적 문제로 작용하였으며 이의 해결을 위하여 사회 전반에 걸쳐 어려움을 겪은 경험을 가지고 있다. 이렇듯 소프트웨어에 대한 중요성이 증가함과 동시에 산업분야에 있어서 산업용 장비에 내장되는 소프트웨어의 품질에 대한 측정 및 평가 그리고 이를 통한 품질향상의 의식들이 고취

† 종신회원 : 호서대학교 벤처전문대학원 교수  
 †† 준회원 : 호서대학교 벤처전문대학원 컴퓨터응용기술학과  
 ††† 정회원 : 산업자원부 기술표준원 공업연구관  
 논문접수 : 2003년 5월 29일, 심사완료 : 2003년 6월 25일

되고 있으며 이와 관련된 활동들이 전개되고 있다. 이를 위하여 선행적으로 이루어져야 할 것은 양질의 소프트웨어에 대한 평가와 인증체계에 대한 구축이라 할 수 있는데[9] 이를 통하여 소프트웨어의 품질을 향상시키는 것은 물론이고 국내적인 시각에서 보면 국내의 소프트웨어 산업에 있어 국제적 수준의 경쟁력 있는 소프트웨어 품질을 확보할 수 있다는 것이다[10]. 이에 따라 현재 국내에서도 국제표준에 준하는 품질인증 절차 및 체계의 구축이 이루어지고 있다. 그리고 뒤를 이어 이러한 평가 체계를 통한 실질적인 평가 작업을 수행할 수 있는 도구의 개발이 이루어져야 할 것이다. 현재 국내의 실정을 놓고 보면 아직 이러한 평가체계가 환경, 기타 제반사항들이 미흡한 실정이며 그의 수행을 위한 도구 또한 많이 부족하다 볼 수 있다[11, 13].

따라서 본 논문에서는 소프트웨어 품질을 측정, 평가하기 위하여 소프트웨어 일반적 품질 요구사항을 위한 국제표준인 ISO/IEC 12119, 소프트웨어 품질특성 및 부특성 평가를 위한 국제표준인 ISO/IEC 9126을 바탕으로 평가 메트릭을 구축하였고, 이의 수행을 위하여 평가모듈의 구성을 위한 국제표준인 ISO/IEC 14598-6을 바탕으로 산업용 소프트웨어의 정량적인 품질을 측정, 평가하여 품질측정의 결과를 토대로 품질을 인증하는 산업용 소프트웨어 국제표준 적합성 인증 시스템을 설계 및 구현하였다.

## 2. 관련 연구

### 2.1 관련 국제품질표준

사회 전반에 걸쳐 소프트웨어 응용부문의 확산과 더불어 기존의 소프트웨어 제품의 개발 공정과 품질 관리를 위한 표준화된 평가절차의 부재로 인한 소프트웨어의 대외적인 신뢰도 저하의 문제점을 개선하기 위하여 ISO(International Standardization Organization) 국제 표준화기구에서는 1987년도부터 JTC1/SC7(Software Engineering) 기술위원회를 설립하여 소프트웨어 공학분야에 대한 국제표준을 제정하기 시작했다[1]. 그 중에서 작업그룹 6(WG6)에서는 소프트웨어 제품평가와 관련한 표준들을 제정해 오고 있다. 본 논문에서 사용된 주요 국제표준으로는 소프트웨어 일반적 품질 요구사항을 위한 국제표준인 ISO/IEC 12119, 소프트웨어 품질특성 및 부특성 평가를 위한 국제표준인 ISO/IEC 9126, 평가 프로세스를 위한 국제표준인 ISO/IEC 14598 등이 있다.

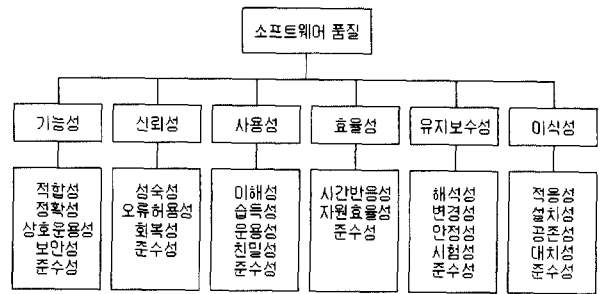
#### 2.1.1 소프트웨어 일반적 품질 요구사항(ISO/IEC 12119)

소프트웨어의 일반적 품질 요구사항에 대한 국제표준인 ISO/IEC 12119는 제품소개를 위한 설명서의 품질적합 여부를 판별하는 제품설명서와 제품사용자를 위한 매뉴얼 내용의 품질적합여부를 판별하는 사용자문서 그리고 기능, 성능 및 범위값 오류방지 등의 품질적합 여부를 판별하는 실행 프로그램의 3가지 품질특성으로 나누어 볼 수 있다[4].

#### 2.1.2 소프트웨어 품질특성 및 부특성(ISO/IEC 9126)

소프트웨어의 품질특성 및 부특성에 대한 국제표준인 ISO/IEC 9126은 부특성을 나타내는 ISO/IEC 9126-1, 외부 메트릭에 대한 ISO/IEC 9126-2, 내부 메트릭에 대한 ISO/IEC 9126-3, 사용자 품질 메트릭에 대한 ISO/IEC 9126-4 로 세분화된다. ISO/IEC 9126-1에서는 소프트웨어 품질을 (그림 1)과 같이 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성의 여섯 가지로 규정하고 있으며 이러한 특성들은 다시 세부적인 부특성들로 구성된다[5].

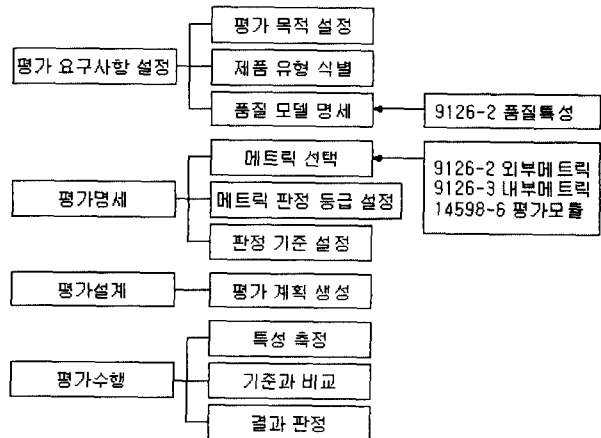
부특성은 ISO/IEC 9126-3의 내부 메트릭이나 ISO/IEC 9126-2의 외부 메트릭에 의하여 정량적인 품질측정이 가능해진다. 품질특성과 부특성은 소프트웨어품질에 대한 일관성 있는 체계를 제공하며, 소프트웨어에 대한 품질 요구사항을 상세화하거나 또는 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성과 같은 소프트웨어 품질특성간의 균형을 맞추기 위한 기본적인 체계를 제공하고 있다. 소프트웨어 품질 부특성은 내부 메트릭이나 외부 메트릭에 의해 측정 가능하다.



(그림 1) 소프트웨어 품질특성과 부특성의 체계

#### 2.1.3 평가자를 위한 평가프로세스(ISO/IEC 14598-5)

ISO/IEC 14598은 6부분으로 구성되어 있으며 전반적인 품질평가 프로세스는 (그림 2)와 같은 절차에 따라 수행되며 ISO/IEC 14598의 부분 3(개발자를 위한 프로세스), 4(구매택



(그림 2) ISO/IEC 14598의 평가 프로세스

를 위한 프로세스), 5(평가자를 위한 프로세스)에서 각 과정의 특성에 따라 변경될 수 있다[3]. 이러한 평가프로세스는 여러 집단이 평가 결과에 대한 이해, 승인 및 신뢰를 필요로 할 때, 소프트웨어 제품평가의 실제적 구현을 위한 요구사항과 권장사항을 규정한다. 특히, 이 평가프로세스는 ISO/IEC 9126에 기술된 품질개념을 적용하기 위해 사용할 수 있다.

(그림 3)은 평가자를 위한 평가프로세스를 나타낸 것으로 5단계의 작업을 수행하여 이루어진다. 첫째로 평가 요구사항을 설정하여 평가의 목적을 기술하게 되고 이를 통하여 서로 다른 관점에서의 제품 사용자들의 요구를 고려할 수 있다. 둘째 평가 명세에서는 평가대상 제품에 대해 수행될 평가와 측정의 범위를 정의하는 단계로 평가를 위해 제출된 제품이나 그 구성 요소들에 대하여 수행될 평가와 측정의 범위를 정의할 수 있다. 이때 주의해야할 사항은 평가명세를 정의하는데 있어 평가자의 개인적인 정보를 반영시키지 말아야 한다[12]. 셋째 평가 설계에서는 평가 명세에 따라 수행하기 위한 절차를 기술하게 되고 평가계획을 생성하고 평가방법을 정의하게 된다. 넷째 평가 수행에서는 평가 요구사항에 따라 소프트웨어 제품을 측정, 검증하는 활동을 수행하여 결과를 도출하는 단계로 평가 수행 과정의 활동을 나타내고 있다. 마지막으로 다섯 번째 평가 결론에서는 평가 보고서 검토 및 평가 데이터 처분에 대해 검토하는 단계로 평가 결론 단계에서 수행하는 활동을 나타내고 있다.



(그림 3) ISO/IEC 14598-5의 평가자를 위한 평가 프로세스

2.1.4 평가모듈 구성(ISO/IEC 14598-6)

소프트웨어 품질특성값의 측정은 측정대상 소프트웨어에 부합되는 시험방법, 절차, 조건 및 결과해석 등에 따라 결과값이 달라지게 된다. 때문에 각각의 메트릭은 동일한 결과값을 얻기 위하여 동일한 환경과 절차에 따라 측정되어야 한다. 평가모듈은 메트릭의 측정환경, 절차 및 기준 등을 정형화하여 측정결과의 반복성, 재현성, 객관성을 확보하고 평가메트릭의 재사용을 용이하게 하기 위하여 각 메트릭별로 평가에 필요한 모든 사항을 통합하여 정의하여야 한다. 이에 따라 ISO/IEC 14598-6에 의해 정의된 평가모듈은 6개의 기본항목과 1개의 선택적 항목으로 구성되어 있으며 <표 1>과 같이 나타내어 볼 수 있다.

<표 1> ISO/IEC 14598-6의 평가 모듈

구 분	항 목	내 용
• EM0 개요	서론 및 소개	모듈목적, 개념, 배경설명.
• EM1 범위	품질특성, 평가수준, 적용기술, 적용범위	적용대상, 품질특성, 부특성, 적용기술 및 적용범위 등을 기술
• EM2 참조규격	관련규격	평가모듈과 관련된 기술
• EM3 용어 및 정의	용어정의	평가모듈에 사용된 용어 정의
• EM4 입력요소와 메트릭	입력요소정의, 데이터요소, 메트릭과 정의	평가에 이용되는 입력요소의 구분, 입력값의 상세 설명, 메트릭 계산식
• EM5 결과해석	결과판정, 결과보고서	판정가능한 값으로 변환, 평가결과 보고서
• EMA 적용절차	소요자원, 평가지침	소프트웨어, 하드웨어, 도구, 기술 및 지식 등 시료선정, 결과계산 등 최종기록사항

2.2 웹 애플리케이션 프레임워크(Web Application Framework)

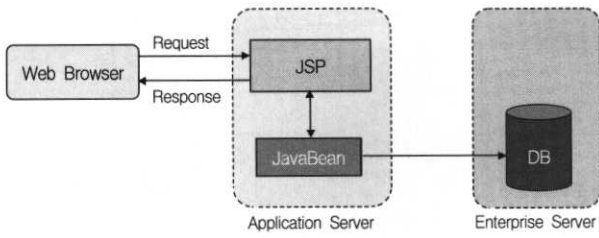
프레임워크(Framework)란 특정형태의 소프트웨어 문제를 해결하기 위해 상호 협력하는 클래스들과 인터페이스들의 집합이라 할 수 있으며 아래와 같은 특성을 가지고 있다[6].

- 특정 개념들의 추상화를 제공하는 여러 클래스나 컴포넌트로 구성된다.
- 추상적인 개념과 같은 문제를 해결하기 위한 작업 방법을 정의한다.
- 컴포넌트들의 재사용이 가능하다.
- 높은 수준의 패턴들을 조직화한다.

이러한 프레임워크는 애플리케이션에서 특정 기능들을 제공하기 위해 확장할 수 있는 일반적이고 상호 협력적인 컴포넌트들을 제공한다.

2.2.1 JSP 모델(Model) 1 아키텍처(Architecture)

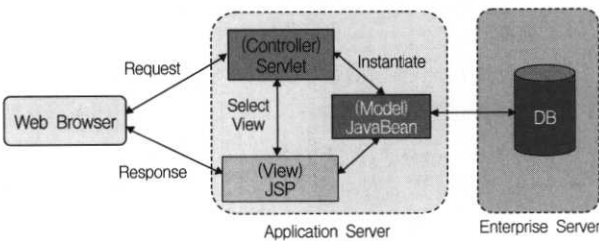
모델 1 구조는 JSP 페이지와 빈, 비즈니스 객체를 사용하며 (그림 4)와 같이 나타낼 수 있다. 모델 1 구조는 요청(request)을 JSP 페이지(Page)로 전달하고 빈(Bean)을 사용하여 간접적으로 비즈니스 객체(Business Object)를 사용한다. 이 방법은 빈과 JSP 페이지를 분리하여 작성하기 때문에 큰 프로젝트를 작성할 때 자주 발생하는 비즈니스 객체의 변경으로 인한 영향을 빈 내부에서만 처리하면 되는 방식이다. 빈의 내용만 변경하고 빈의 인터페이스가 변경되지 않는 한, JSP 페이지는 빈의 비즈니스 로직의 구현 내용과 무관하게 작성될 수 있다. 그러나 이러한 모델 1 구조에서는 JSP 페이지에서 내용표현과 내용 생성부분을 처리할 때 자바 코드와 관련하여 수행되는 부분이 있기 때문에 비즈니스 객체 작성과 웹 페이지 작성을 분리하여 작업하는데 어려움이 있다.



(그림 4) JSP Model 1 Architecture

2.2.2 JSP 모델(Model) 2 아키텍처(Architecture)

모델 2 구조는 모델 1 구조와 같이 JSP 페이지와 비즈니스 객체를 분리하여 프로젝트를 작성하는 것을 기본적인 내용으로 한다. 개발 중에 비즈니스 객체의 상태가 변경될 요소가 많은 경우에는 비즈니스 객체의 개발이 JSP 페이지와 연관되지 않도록 하기 위해서 모델 2 구조를 사용하는 것이 바람직하게 된다. 이러한 모델 2 구조는 (그림 5)와 같이 나타낼 수 있으며 내용을 표현하는 부분과 생성하는 부분이 분리되어 처리되는 것을 볼 수 있다.



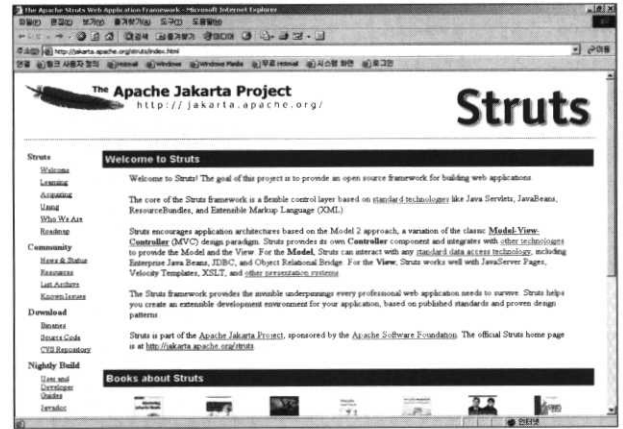
(그림 5) JSP Model 2 Architecture

이러한 모델 2 구조는 Smalltalk에 기초를 두고 있는 MVC (Model-View-Controller) 패턴(Pattern)을 기반으로 하며 MVC의 관점에서 보면 모델(Model) 부분이 비즈니스 객체에 해당하고, 뷰(View) 부분이 JSP 페이지, 그리고 컨트롤러(Controller) 부분이 서블릿(Servlet)에 해당한다 볼 수 있다. 이러한 MVC 구조는 객체 추상화 작업을 통하여 캡슐화하고, 클래스를 작성함으로써 객체간의 결합도(Coupling)를 낮추어 의존도를 낮추고 재사용성을 증대시키는 효과를 가져온다.

2.2.3 아파치 스트러츠 프레임워크(Apache Struts Framework)

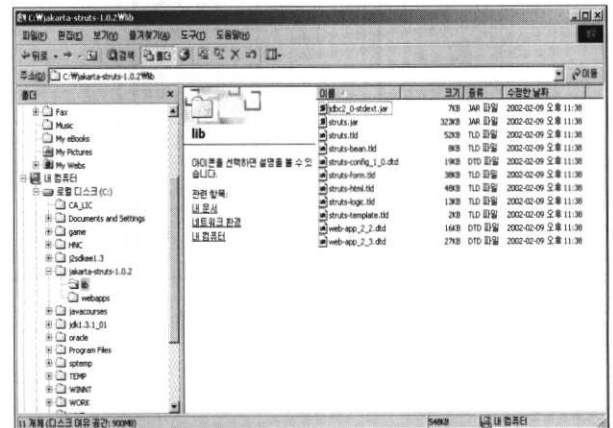
스트러츠 프레임워크(Struts Framework)는 자바 서블릿(Java Servlet)과 자바서버 페이지(JSP, JavaServer Pages) 기술에 기반한 웹 애플리케이션을 더 쉽게 만들 수 있도록 개발된 오픈 소스(Open Source) 프레임워크이다[7]. 스트러츠 프레임워크는 개발자들이 웹 애플리케이션을 구축할 때 기반으로 삼을 수 있는 통합 하부 구조물로서 실제 개발에 있어서 프로그래머가 비즈니스 애플리케이션 구현에 집중할 수 있도록 한다. 이러한 스트러츠 프레임워크는 모델(Model) 2 방식을 기반으로 작성된 것으로 크레이그 맥클래너한(Craig McClanahan)이 만들어 2000년 아파치 소프트웨어재단(ASF,

Apache Software Foundation)에 기증한 것으로 현재 전 세계에 걸쳐 많은 개발자들에 의해 사용되고 있다. 이와 관련된 자세한 내용은 (그림 6)과 같이 <http://jakarta.apache.org/struts/index.html>에서 찾아볼 수 있다.



(그림 6) Struts Framework 관련 Site

스트러츠를 설치하기 위해서는 위의 사이트에서 소스 또는 바이너리를 다운로드하여 설치하여야 한다. 현재 제공하고 있는 버전으로는 1.0.2로 설치된 모습은 (그림 7)과 같으며 스트러츠를 사용하기 위해서는 설치폴더의 lib 폴더안에 있는 struts.jar 파일을 해당 애플리케이션의 클래스패스(classpath)에 추가하면 된다.



(그림 7) struts 설치화면

3. 산업용 소프트웨어 국제표준 적합성 인증 시스템

3.1 적합성 인증 절차

산업용 소프트웨어에 대한 국제표준 적합성 인증 절차는 크게 7단계의 작업으로 구성될 수 있다. 신청 및 접수를 통하여 해당 산업용 소프트웨어에 대한 인증 수행이 적합한지를 선별하고 그것에 대한 평가계획을 통하여 사용성 평가를 위한 시험 정보를 추출·선정 하게 되고 적합성 기술

위원회를 거쳐 품질특성별 메트릭을 선정하고 인증기준안을 심의·제정하게 된다. 그리고 이를 토대로 현장실사작업을 통하여 개발환경 및 품질관리의 실시 여부 등을 조사하고 나서 품질시험 평가작업을 통해 적합성기술 위원회에서 제정한 기준에 의거하여 평가 작업을 수행하고 이를 통하여 얻은 결과를 적합성 인증위원회에 회부하여 평가결과를 심의하게 된다. 이러한 심의를 거쳐 평가 기준에 통과한 제품에 대하여 인증서를 수여하게 되는데[16] 이러한 내용은 <표 2>에서는 같이 정리해 볼 수 있다.

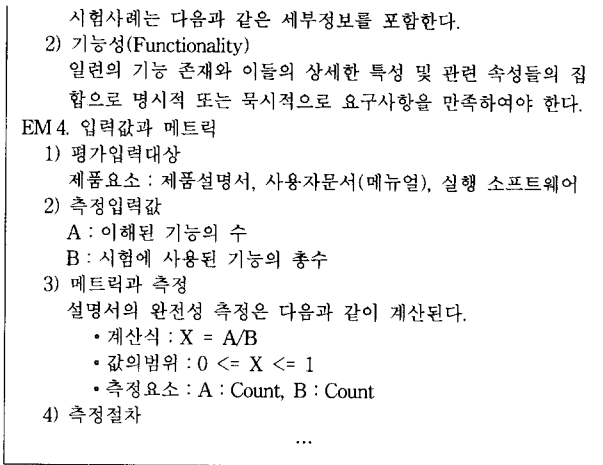
<표 2> 산업용 소프트웨어 국제표준 적합성 인증 절차

인증 절차	내용
신청 및 접수	신청서 및 관련서류 접수, 검토보완
사전조사	신청업체 방문, 개발조직, 인력 및 시험환경등 조사
인증기준안 작성	소프트웨어 분야별 인증기준(안) 및 기능별 시험계획서 작성
적합성기술위원회 (1차회의)	소프트웨어 제품별 인증기준(안) 심의 및 인증기준 제정
현장실사	신청제품의 현장운영 실태조사 및 전문가 현장지도
제품평가수행	인증기준에 의거 제품평가(현장시험 병행)
적합성인증위원회 (2차회의)	평가결과의 적정성 검토 및 인증 적합여부 심의
시험인증서 수여	우수품질 제품에 시험인증서 수여

3.2 평가 모듈의 구성

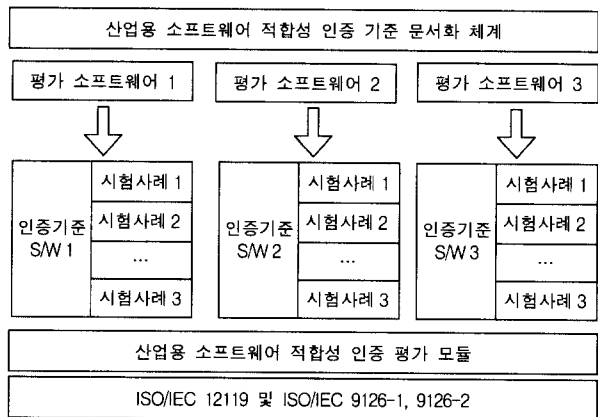
산업용 소프트웨어 국제표준 적합성 인증의 평가모듈은 평가모듈에 관한 국제표준인 ISO/IEC 14598-6을 기반으로 구성하였다. 각각의 단계는 서론 및 소개를 통하여 EM0를 나타내고 EM1은 적용범위, EM2는 관련규격, EM3는 용어 및 정의, EM4는 입력값과 메트릭, EM5는 결과분석, EM6는 적용절차와 같은 절차를 제정하여 각각의 메트릭에 대한 평가모듈을 구성하게 된다. 이에 대한 예는 (그림 8)과 같이 나타내 볼 수 있으며 이 예는 품질특성 중 사용성에 대한 이해성 중 설명서의 완전성에 관한 평가모듈의 일부분이다.

EM No. 6.1.1	이해성-설명서의 완전성
EM 0. 서론 및 소개	
1) 서론	<ul style="list-style-type: none"> <li>• 작성자 : 작성부서 및 성명</li> <li>• 소프트웨어의 국제표준 적합성평가를 위한 사용성 평가모듈 제정</li> </ul>
2) 소개	<ul style="list-style-type: none"> <li>• 규정된 조건에서 사용될 경우, 사용자에게 의해 이해되고, 학습되고, 사용되고 선호될 수 있는 소프트웨어 제품의 능력</li> </ul>
EM 2. 관련규격	ISO/IEC 9126-2 Information Technology-Product Evaluation -Quality and Metric
EM 3. 용어 및 정의	
1) 시험사례(Test Case)	<p>시험자를 위한 문서화된 지침으로써, 어떤 기능이나 기능의 조합이 어떻게 시험되어야 하는가를 명시하고 있다. 하나의</p>



(그림 8) 평가모듈 구성의 예

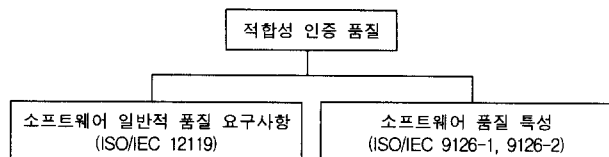
그리고 이러한 평가모듈의 문서화의 체계는 소프트웨어 분야별로 적용 가능한 시험항목 및 인증요건 등을 규정하는 문서로써 소프트웨어 분야별로 적합성 인증위원회의 심의를 거쳐 제정되며 (그림 9)와 같이 나타낼 수 있다.



(그림 9) 인증기준의 문서화 체계

3.3 품질 평가 항목

산업용 소프트웨어 품질 평가 항목의 구축은 두 가지 국제 표준에 기인하여 구축하였는데 (그림 10)과 같이 소프트웨어의 일반적 품질요구사항에 대한 국제표준인 ISO/IEC 12119와 소프트웨어 품질특성 및 부특성 평가에 대한 국제표준인 ISO/IEC 9126가 그것이며 전체 145개 항목으로 이루어져 있다[15]. 그리고 이에 대한 설명은 <표 3>과 같이 정의해 볼 수 있다.



(그림 10) 적합성 인증을 위한 품질모델

<표 3> 품질특성별 측정항목

구 분	품질특성별	내 용	항목수
일반적 품질요구사항 ISO/IEC 12119	제품설명서	제품설명서에 관한 품질요구사항	22
	사용자문서	사용자문서에 관한 품질요구사항	10
	실행 소프트웨어	실행소프트웨어에 관한 품질요구 사항	12
소프트웨어 제품품질특성 ISO/IEC 9126-2	기 능 성	요구되는 기능을 제공하는 능력	12
	신뢰성	지정된 수준의 성능을 유지하는 능력	17
	사 용 성	쉽게 이해하고 배울 수 있게 하는 능력	25
	효 율 성	적은 자원으로 적절한 성능을 제 공하는 능력	22
	유지보수성	개선·수정에 쉽게 대응할 수 있 는 능력	15
	이 식 성	다른 환경(소프트웨어, 하드웨어) 에서 운영될 수 있는 능력	10
Total			145

메 트릭 명	세 부 항 목	계산식	값범위
4.1.1 기능 적정성	A	평가 중에 문제점이 내포된 기능의 수	1-(A/B) 0~1
	B	시험에 사용된 기능의 수	
4.1.2 기능구현의 완전성	A	평가 중 확인된 결여 기능 의 수	1-(A/B) 0~1
	B	요구명세서에 기술된 모든 기능의 수	
4.1.3 기능구현 정도	A	평가 중에 확인된 결여기능 과 잘못 구현된 기능의 수	1-(A/B) 0~1
	B	요구명세서에 기술된 모든 기능의 수	
4.1.4 기능명세의 안전성	A	운영시작 후 변경된 기능의 수	1-(A/B) 0~1
	B	요구명세서에 기술된 모든 기능의 수	

(그림 11) 메트릭의 예

그리고 이러한 품질특성별 메트릭의 예는 (그림 11)과 같  
이 구성할 수 있으며 이것은 ISO/IEC 9126-2를 기반으로 작  
성된 품질특성 중 기능성에서 적합성에 관한 메트릭을 나  
타낸 것이다.

3.4 가중치 및 품질특성 계산

품질 특성별 가중치는 <표 4>와 같은데 이는 특성별 세  
부 메트릭들의 측정값을 토대로 품질특성별로 산업용 소프  
트웨어의 제품 특성에 따라 부여한 것이다. 이 가중값은 제  
품에 따라 조정이 가능하며 <표 4>에 나타난 값은 일반적  
으로 부여된 기본값이라 할 수 있다. 그리고 품질 특성별 품  
질측정이 이루어진 것을 바탕으로 해당 제품의 품질을 측정  
할 수 있는데 이는 먼저 품질 주특성별 결과값을 도출하여  
이들의 전체 합으로 이루어진다. 이와 같은 계산식과 측정  
항목 등에 대해서는 (그림 12)와 같다.

<표 4> 품질 주특성별 가중값

품질 주특성	품질중요도	가 중 값
기 능 성	H	0.3
신뢰성	H	0.3
사 용 성	M	0.2
효 율 성	L	0.1
유지보수성	L	0.05
이 식 성	L	0.05

계 산 식	
$Q(i) = \sum (MV(i, j)) / N(i) \times W(i) \times 100$	
$Q_{tot} = \sum (Q(i))$	
측정요소	설 명
i	품질 주특성
j	품질 주특성에 포함되어 시험된 메트릭
MV(i, j)	측정된 각 품질특성별 메트릭 측정 결과값
N(i)	품질특성별로 시험에 적용된 메트릭 수
W(i)	품질 주특성별 가중값
Q(i)	품질 주특성별 결과값
Q <sub>tot</sub>	소프트웨어 총 품질 결과값

(그림 12) 품질특성 계산식

그리고 계산된 품질값에 대한 품질평가 등급의 기준은 <표  
5>와 같이 이루어지며 이를 통하여 신청제품에 대한 품질  
인증을 판별하는 것이다. 통상적으로 품질인증의 적합한 등  
급은 A등급이며 경우에 따라서 등급값의 변형이 이루어질  
수도 있다.

<표 5> 품질평가 등급

번 호	총 품질 계산식(Q <sub>tot</sub> )	평가등급
1	90이상~100	A
2	80이상~90	B
3	70이상~80	C
4	60이상~70	D
5	0~60미만	E

4. 시스템 구현

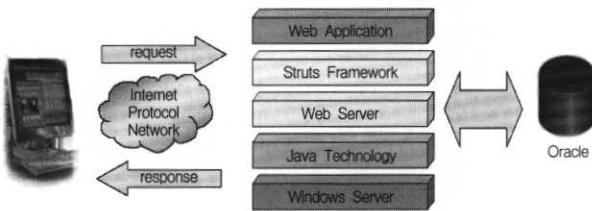
산업용 소프트웨어 국제표준 적합성 인증 시스템의 구현  
은 웹 환경을 기반으로 하는 웹 애플리케이션의 형태를 띠  
고 있다. 이는 사용자로 하여금 접근성과 편의성 그리고 기  
타 웹이 보유하고 있는 많은 장점들을 가져다 줄 수 있다  
[14]. 이러한 웹 환경에 웹 애플리케이션 프레임워크인 스트  
러츠(Struts)를 탑재하여 개발의 생산성 및 유지보수성을  
향상시켰고 그 위에 자바 기술을 사용하여 객체지향 기술  
을 이용할 수 있었다. 실제 구현작업에 있어서도 기존의 개  
발방법에 비해 개발기간이 크게 단축되었고 개발 후에 기  
능의 추가 또는 수정 심지어는 설계의 변경에 따른 수정작

업 초차도 기존의 방법보다 쉽고 빠르게 수행할 수 있었다.

4.1 시스템 구성 및 개발환경

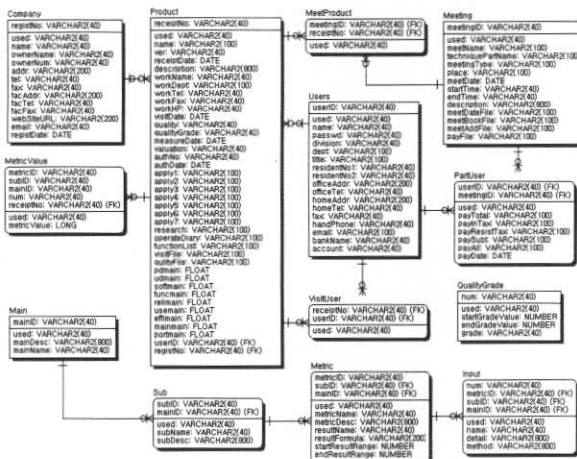
본 시스템은 개발 후 설치되는 운영체제의 기종이 Windows 2000 Server이기 때문에 그와 유사한 운영체제이며 상호호환성을 유지하는 운영체제인 Windows 2000 Professional을 사용하였다. 개발언어로는 자바(Java) 기술을 사용하였다. Java SDK의 버전으로는 1.3.1을 사용하였고 J2EE는 1.3 버전을 사용하였다. 그리고 웹을 통해 작동될 수 있도록 JSP (Java ServerPages) 기술을 사용하였다. 데이터베이스(Database)는 오라클(Oracle)사의 버전 9를 사용하였고, 웹 서버는 Apache Tomcat 4.1을 그리고 실제 개발에서 유용하게 사용된 웹 애플리케이션 프레임워크인 스트러츠의 버전은 1.0.2를 사용하였다. 이와 같은 시스템의 구성은 (그림 13)과 같이 나타낼 수 있고 위의 내용은 아래와 같이 정리해 볼 수 있다.

- 운영체제 : Windows 2000 Professional
- 개발언어 : Java SDK 1.3.1, J2EE 1.3, JSP 1.2
- 데이터베이스 : Oracle 9i
- 웹 서버 : Apache Tomcat 4.1
- 프레임워크 : Apache Jakarta Struts 1.0.2



(그림 13) 시스템 구성도

4.2 데이터베이스 설계



(그림 14) DB 설계

데이터베이스는 전체 13개의 테이블로 구성되어 있다. 주

요 테이블로는 신청 제품과 관련된 PRODUCT 테이블과 신청제품을 의뢰한 COMPANY 테이블 그리고 인증위원 및 제품의 시험을 담당하게 되는 USERS 테이블이 있고 매트릭과 관련된 테이블들이 있다. 매트릭과 관련된 테이블은 전체 4개의 테이블로 구성이 될 수 있는데, 주특성을 나타내는 MAIN 테이블, 부특성을 나타내는 SUB 테이블 그리고 매트릭의 일반정보를 나타내는 METRIC 테이블과 각각의 매트릭을 가지고 품질측정 작업을 수행하기 위한 데이터가 담겨있는 INPUT 테이블이 그것이라 할 수 있다. 이와 같은 전체 테이블의 구조는 (그림 14)와 같은 관계로 구성되어 있음을 볼 수 있다.

그리고 이러한 데이터베이스의 용도는 <표 6>과 같이 정리해 볼 수 있다.

<표 6> 데이터베이스 Table

테이블명	설 명
Company	인증의뢰 업체 관련 테이블
Product	인증의뢰 제품 관련 테이블
Users	시스템에서 사용되는 사용자 관련 테이블 (관리자, 사용자, 업체담당자, 인증위원)
MeetProduct	회의에 부결된 제품을 관리하는 테이블
Meeting	회의 관련 테이블
PartUser	회의참석인증위원 관련 테이블
VisitUser	현지실사참여 인증위원 관련 테이블
MetricValue	추출된 매트릭 측정값 관련 테이블
Main	품질 주특성 관련 테이블
Sub	품질 부특성 관련 테이블
Metric	품질 매트릭 관련 테이블
Input	품질 매트릭 입력치 관련 테이블
QualityGrade	품질 평가기준 관련 테이블

4.3 기능 모듈

시스템의 기능은 크게 5가지의 기능으로 구분해 볼 수 있다. 업체등록·검색·수정, 제품등록·검색·수정, 사용자 등록·검색·수정, 회의등록·검색·수정, 제품시험이 그것이며 이 중에서 회의관련 기능과 제품시험은 다시 세부기능들의 조합으로 이루어진다. 회의등록은 전체 5가지의 순차적인 단계로 이루어지는데 일반정보등록, 회의제품선택, 회의참석위원선택, 회의관련 파일등록, 전체등록정보 확인의 순서로 이루어진다. 이는 실제 업무의 수행에 있어서 회의업무는 하나의 제품에 종속적이지 않고 별도로 관리되기 때문이다. 그리고 회의 수정도 등록과 같은 절차로 이루어지게 된다. 제품시험은 실제 제품의 품질을 측정하기 위한 전체 프로세스를 기반으로 설계되었기 때문에 6단계의 세부기능으로 구성된다. 신청서 접수에 대한 관리, 사전조사서에 대한 관리, 인증기준안작성과 관련된 작업, 현장실사작업 관리, 제품시험 그리고 인증결과 출력의 기능들이 그것이라 할 수 있다. 제품시험과 관련된 세부 기능들은 <표

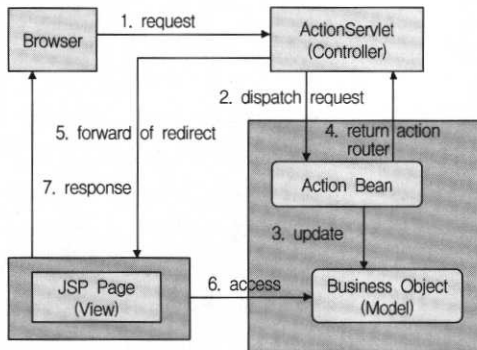
7>과 같이 정리해 볼 수 있다.

<표 7> 제품시험의 세부 기능

제품시험 기능	제품 시험 세부 기능	설 명
신청서접수	신청서 접수 현황	신청서 접수시 등록된 정보를 보여줌
	신청서 접수 관련 파일	신청서 접수시 제출된 관련문서 관리
사전조사서	사전 조사 정보	사전조사 정보를 등록, 관리
	사전조사 관련파일	사전조사서 작성된 문서 관리
인증 기준안 작성	기준안선택	제품 시험을 위한 기준안 선택
	기준안확인	선택된 기준안 확인
현장 실사	현장실사 참석위원	현장실사에 참석하는 위원 선택, 관리
	현장실사 관련파일	현장실사시 작성된 문서 관리
제품 시험	가중치 입력	제품시험에 사용되는 제품의 가중치 입력
	측정값 입력	제품시험을 위한 기준안의 각각의 메트릭에 대한 측정치 입력
인증 결과	제품 시험결과 및 인증결과	제품의 품질값을 확인하고 그에 따른 인증결과 확인

4.4 구 현

본 시스템의 구현은 (그림 15)의 스트러츠 프레임워크(Struts Framework) 구조와 같이 화면의 프리젠테이션 부분을 담당하는 JSP Page와 이벤트를 처리하는 서블릿(Servlet) 그리고 비즈니스 로직을 담당하는 ActionBean과 Model로써 데이터를 가지고 있는 FormBean들로 구분해 볼 수 있다. 이는 JSP 화면에서 이벤트를 발생시키면 그 이벤트에서 사용되는 데이터들은 FormBean이 담당하고 내부적인 처리는 ActionBean들이 FormBean을 호출함으로 해서 해당 데이터들을 처리한 후 처리한 내용을 화면에 전달하는 것이다[8].



(그림 15) Struts Framework

4.4.1 시스템 설정

스트러츠 프레임워크는 애플리케이션에 필요한 컴포넌트들을 적재하기 위해서 설정 사항들을 정의하여 선언하는 방

식의 설정파일을 제공한다. 설정파일은 XML 형식으로 이에 대한 정의는 (그림 16)과 같다.

```

<?xml version = "1.0" encoding = "ISO-8859-1" ?>
<!DOCTYPE struts-config PUBLIC
    "-//Apache Software Foundation//DTD Struts
    Configuration 1.0//EN"
    "http://jakarta.apache.org/struts/dtds/struts-config_
    1_0.dtd" >
<struts-config >
<!-- ===== Data Source Configuration ===== -->
<data-sources >
<data-source >
    autoCommit = "false"
    description = "AFTS auth tool DataBase -Oracle 9i"
    driverClass = "oracle.jdbc.driver.OracleDriver"
    maxCount = "4"
    minCount = "2"
    password = "####"
    url = "jdbc : oracle : thin : @211.192.249.### : 152
    1 : ###"
    user = "####"
    />
</data-sources >
<!-- ===== Form Bean Definitions ===== -->
<form-beans >
<form-bean name = "productForm"
    type = "kr.co.microjava.afts.entity.ProductForm"/>
</form-beans >
<!-- ===== Global Forward Definitions ===== -->
<global-forwards >
<forward name = "logon"
    path = "/jsp/logon.jsp"/>
</global-forwards >
<!-- ===== Action Mapping Definitions ===== -->
<action-mappings >
<action path = "/product"
    type = "kr.co.microjava.afts.action.ProductAction"
    name = "productForm"
    scope = "request" >
</action >
<action path = "/company"
    type = "kr.co.microjava.afts.action.CompanyAction"
    name = "companyForm"
    scope = "request" >
</action >
</struts-config >
    
```

(그림 16) struts-config.xml 설정파일의 일부

그리고 시스템 동작에 있어서 서블릿 호출을 위한 기본 설정을 수행하여야 하는데 이와 같은 설정은 (그림 17)과 같다.

```

<?xml version = "1.0" encoding = "ISO-8859-1"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
    2.2//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd" >
<web-app >
<servlet >
<servlet-name > action </servlet-name >
    
```



```

< servlet-class > org.apache.struts.action.ActionServlet
</servlet-class >
< init-param >
  < param-name > application </param-name >
  < param-value > ApplicationResources </param-value >
</init-param >
< init-param >
  < param-name > config </param-name >
  < param-value > /WEB-INF/struts-config.xml
  </param-value >
</init-param >
< init-param >
  < param-name > detail </param-name >
  < param-value > 2 </param-value >
</init-param >
< load-on-startup > 2 </load-on-startup >
</servlet >
<!-- Standard Action Servlet Mapping -->
< servlet-mapping >
  < servlet-name > action </servlet-name >
  < url-pattern > *.do </url-pattern >
</servlet-mapping >
<!-- The Usual Welcome File List -->
< welcome-file-list >
  < welcome-file > index.jsp </welcome-file >
</welcome-file-list >
<!-- Struts Tag Library Descriptors -->
< taglib >
  < taglib-uri > /WEB-INF/struts-bean.tld </taglib-uri >
  < taglib-location > /WEB-INF/struts-bean.tld
  </taglib-location >
</taglib >
</web-app >

```

(그림 17) web.xml 설정파일

위와 같이 servlet 요소내에 있는 servlet-name 요소의 값이 servlet-mapping 요소에 있는 servlet-name 요소의 값과 일치해야 하고, 이것은 웹 서버가 확장자가 .do로 입력되는 요청을 ActionServlet으로 서비스하게 되는 것이다.

#### 4.4.2 이벤트 처리

위와 같이 설정작업이 이루어지면 실제 시스템이 동작할 수 있는 기본환경이 구축된 것으로 시스템을 운영, 관리할 수 있다. 시스템의 동작에 있어서는 웹 애플리케이션 프레임워크인 스트러츠의 기본 동작원리에 따라 이벤트를 주고, 받게 되는데 JSP 페이지에서 이벤트를 요청을 하면 컨트롤러인 ActionServlet이 요청한 이벤트를 감지를 하고 실제 작업을 수행하는 ActionBean을 호출하여 작업이 이루어지게 된다. 이때 데이터를 서로 주고, 받는 수단으로는 ActionForm을 이용하게 되는데 ActionForm은 하나의 객체로 데이터의 전달에 있어 각각의 데이터에 대한 처리를 지양하고 객체 단위로 데이터를 담아서 처리하게 된다. 이렇게 서로 이루어지는 요청에 대한 응답은 struts-config.xml 설정파일에서 ActionServlet과 ActionForm의 Mapping에 의하여 각각이 처리해야할 데이터가 무엇인지를 정의하게 된다.

```

< html >
< head >
< title > 산업용 소프트웨어 국제표준 적합성 인증 </title >
< link rel = "stylesheet" href = "../include/ats.css"
  type = "text/css" >
< script language = "JavaScript" >
<!--
function formSubmit ( )
{
  document.searchProduct.action = "/product.do?Action
    = login&page = 1" ;
  document.searchProduct.submit ( ) ;
}
function next ( page )
{
  document.searchProduct.action = "/product.do?Action = login&
  <% = searchString% > &page =" +page ;
  document.searchProduct.submit ( ) ;
}
-->
</script >

```

(그림 18) JSP 화면에서 이벤트 요청의 예

(그림 18)은 JSP 페이지에서 이벤트를 요청하는 예로 ActionServlet을 호출하는데 설정파일에서 정의해 놓은 product.do를 호출하는 모습을 볼 수 있다. ActionServlet을 호출할 때는 Action 파라미터(Parameter)를 같이 전송하게 되며 해당 이벤트를 처리하는 ActionServlet은 이 파라미터 값에 따라 해당 작업을 처리하게 된다. 이와 같은 모습은 (그림 19)와 같다.

```

public final class ProductAction extends Action
{
  /**
   * perform method override
   * client의 http request를 수행하는 메소드
   */
  ....
  public ActionForward perform ( ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response )
    throws IOException, ServletException
  {
    if ( action.equals ( "login" ) )
    {
      StringBuffer str = new StringBuffer ( ) ;
      try
      {
        conn = Util.getConnection ( servlet ) ;
        ProductForm [ ] product = null ;
        Integer totalCnt = null ;
        ....
      }
      return new ActionForward ( "/jsp/mainTable.jsp?page
        = "+page ) ;
    }
  }
}

```

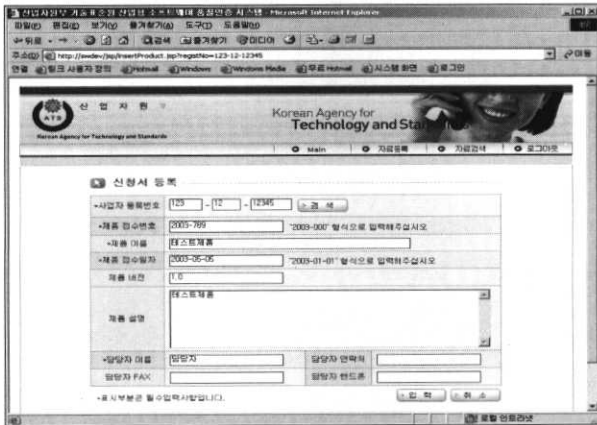
(그림 19) ActionServlet의 예

5. 시험 및 결과

구현된 산업용 소프트웨어 국제표준 적합성 인증 시스템의 테스팅을 위하여 실제 업무와 동일한 작업을 수행해 본 결과 작업을 원활히 수행하기 위해서는 다음과 같은 작업 절차를 따라야 한다. 우선 인증을 의뢰한 업체에 대한 기본 정보를 입력하여야 한다. 이는 하나의 업체에 다수의 제품이 존재할 가능성이 있기 때문이다. 업체정보의 등록을 마치면 신청제품에 대한 정보를 입력하게 된다. 신청제품정보 등록 시 해당 제품에 대한 업체의 사업자등록번호를 입력하게 되는데 이는 이미 등록된 업체데이터를 검색하여 선택하면 자동으로 입력하게 된다. 이와 같은 모습은 (그림 20) 및 (그림 21)과 같다.



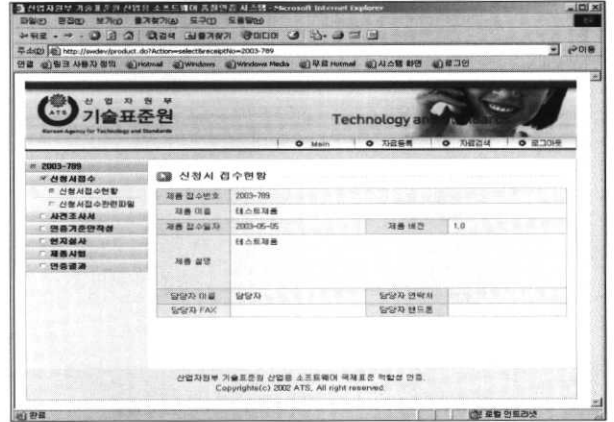
(그림 20) 업체등록



(그림 21) 신청제품의 등록

위와 같이 신청제품의 정보를 입력하고 나면 이제 실제 제품의 품질을 측정하는 작업을 수행할 수 있게된다. 제품의 품질을 측정하는 부분으로 이동하기 위해서는 화면상단의 Main 버튼을 선택하고 선택하여 이동한 화면에서 해당제품의 제품접수번호를 선택하면 이동하게 된다. 그것의 그림은 (그림 22)와 같다. 그리고, (그림 23)은 인증기준을 선택

하는 작업으로 품질특성별로 그리고 각각의 품질특성에 대한 부특성별로 구분하여 선택할 수 있다.

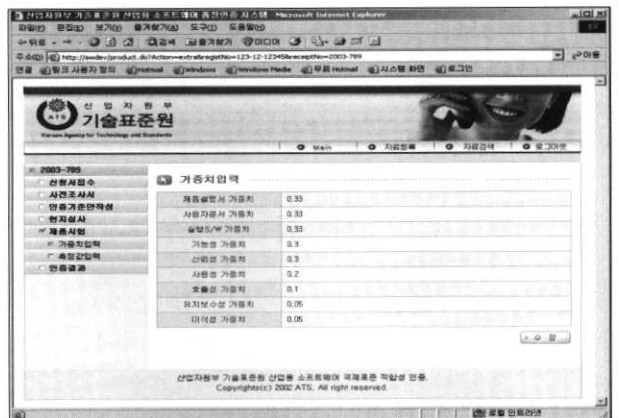


(그림 22) 제품 품질측정

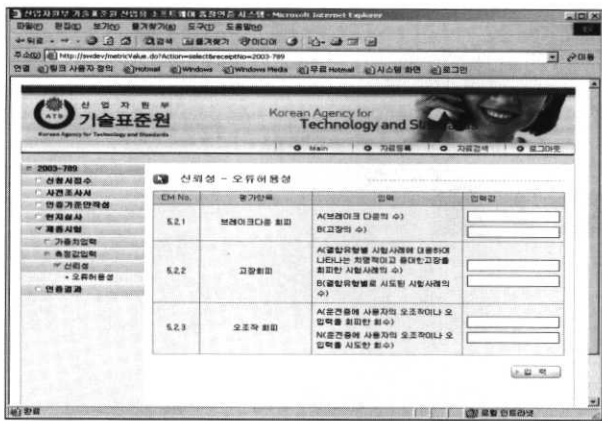


(그림 23) 인증기준 작성

(그림 24)는 신청 제품에 대한 품질특성 가중치를 입력하는 단계로 산업용 소프트웨어에 대한 특성을 반영하여 도출한 것으로 이것은 고유한 제품 본연의 특성에 따라 임의의 수정이 가능하다.

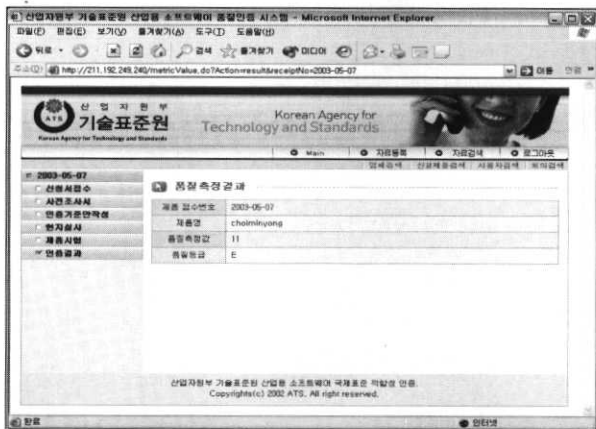


(그림 24) 품질 특성 가중치



(그림 25) 품질 특성값 입력

(그림 25)는 신청제품의 실제 측정된 품질특성 메트릭값을 입력하는 단계로 각각의 메트릭에 대한 입력요소를 입력하게 되면 시스템 내부에서 자동으로 각각의 메트릭의 대한 계산이 이루어지며 또한 이러한 각각의 값들을 취합하여 자동으로 품질 특성값, 품질 결과값 등을 계산하게 된다.



(그림 26) 품질 측정의 결과

(그림 26)은 품질측정 결과를 출력하는 화면으로 (그림 25)에서의 품질특성 메트릭값의 입력 과정을 통하여 품질 계산식에 따라 자동으로 계산이 이루어지게 된다.

## 6. 결 론

산업분야에 있어서 작업을 수행하는 기계장치의 성능향상을 위해서 소프트웨어가 차지하는 중요성이 점점 증가함에 따라 해당 장치에 대한 성능과 품질에 있어서 소프트웨어가 차지하는 중요도 또한 급격히 증가해 지고 있다. 그러나 현재 국내의 실정으로는 이러한 중요한 역할을 담당하고 있는 소프트웨어에 대한 평가환경이 매우 미흡하고 이는 차후 산업분야 전체로의 악영향을 미칠 우려가 있다. 따라서 이러한 소프트웨어에 대한 품질을 평가할 수 있는 환경이 구축되어야 한다. 평가를 위한 평가체계와 평가절차

및 평가모듈 그리고 이러한 기반을 바탕으로 실제작업을 수행하도록 할 수 있는 도구 등 다양한 각도에서의 노력이 요구된다. 이에 본 논문에서는 위와 같은 문제점을 분석하여 이러한 현재 환경의 개선을 위하여 산업용 소프트웨어에 대한 국제표준 적합성 인증 시스템을 설계 및 구현하였다. 소프트웨어에 대한 국제표준을 바탕으로 산업용 소프트웨어에 대한 국제표준 적합성 판별의 기준을 정의하고 웹을 통한 작업이 이루어지도록 하였다.

본 연구 결과는 실질적인 산업용 소프트웨어에 대한 품질평가 및 품질향상을 위해 사용될 수 있다. 또한 산업용 소프트웨어에 대하여 종합적이고 정량적인 품질평가를 통해 고품질의 소프트웨어의 개발을 촉진시킬 수 있으며 객관적인 평가 및 인증이 이루어질 것이다. 향후 연구 과제로는 신청제품의 품질측정결과와 관련하여 좀 더 신뢰성 있는 데이터를 제공해야 할 것이며, 실제 품질측정을 통하여 제품의 품질특성을 나타내며 평가자의 주관적인 견해의 삽입을 미연에 방지할 수 있는 메트릭을 개발하여 산업용 소프트웨어의 특성을 잘 반영할 수 있도록 지속적인 보완과 노력을 기울여야 할 것이다.

## 참 고 문 헌

- [1] Moller, K. H. and Paulish, D. J., "Software Metrics," Chapman & Hall (IEEE Press), 1993.
- [2] Wallmuller, E., "Software Quality Assurance A practical approach," Prentice Hall, 1994.
- [3] ISO/IEC 14598, "Information Technology-Software product evaluation-Part 5, 6," 1997.
- [4] ISO/IEC 12119, "Information technology-Software packages-Quality requirements and testing," 1994.
- [5] ISO/IEC 9126, "Information technology-Software Quality Characteristics and Metrics-Part 1, 2, 3," 1997.
- [6] David M. Geary, Advanced JSP, PHPTR, Sun Microsystem Press, 2001.
- [7] Chuck Cavaness, Programming Jakarta Struts, O'Reilly, Nov., 2002.
- [8] <http://www-106.ibm.com/developerworks/ibm/library/j-struts/>.
- [9] 水野幸男, "소프트웨어의 綜合的品質管理," 日科技連出版, 1993.
- [10] 吉澤. 東. 片山, "소프트웨어의 品質管理と生産技術," 日本規格協會, 1990.
- [11] 양해술, 이하용, "설계단계에서의 품질평가 툴킷(ESCORT-D)의 설계 및 구현", 한국정보과학회논문지(C), Vol.3, No.3, 1997.
- [12] 양해술, "한진해운 신정보(영업 및 물류)시스템의 품질보증과 품질평가", 한진해운(주) 구현단계 확인평가, 1998.
- [13] 양해술, "소프트웨어 제품 평가 지원도구의 개발", ETRI 컴퓨터·소프트웨어 기술연구소 용역과제, 3차년도 최종보고

서, 1999.

- [14] 양해술, 이하용, "소프트웨어 품질평가 도구의 개발", 한국정보과학회 춘계학술발표논문집, 2000.
- [15] 양해술, "Embedded 소프트웨어의 품질평가모델 개발 연구", 한국정보통신기술협회 위탁과제 1차년도 최종보고서, 2002.
- [16] 양해술, "내장형 소프트웨어 품질평가 툴킷 개발", 산업자원부 기술표준원 최종보고서, 2002.



### 양 해 술

e-mail : hsyang@office.hoseo.ac.kr

1975년 홍익대학교 전기공학과(학사)

1878년 성균관대학교 정보처리학과(석사)

1991년 日本 오사카대학 정보공학과 S/W 공학전공(공학박사)

1975년~1979년 육군중앙경리단 전산실 시스템분석 장교

1980년~1995년 강원대학교 전자계산학과 교수

1986년~1987년 日本 오사카대학 객원연구원

1994년~1995년 한국정보처리학회 총무이사, 논문편집위원장

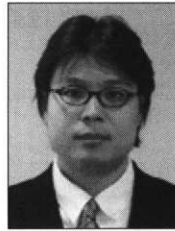
1995년~2002년 한국S/W 품질연구소 소장

1999년~현재 호서대학교 벤처전문대학원 교수

2001년~현재 한국정보처리학회 부회장

2003년~현재 미국 ACIS 학회 Vice President

관심분야 : 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질 관리와 컨설팅, OOA/OOD/OOP, CASE, SI), 프로젝트 관리, CBD기반기술, IT품질경영



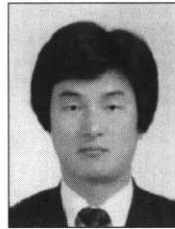
### 최 민 용

e-mail : choiminyong@hotmail.com

1995년~2002년 호서대학교 컴퓨터공학과 (학사)

2002년~현재 호서대학교 벤처전문대학원 컴퓨터응용기술학과 석사과정

관심분야 : 소프트웨어공학(S/W품질평가, 개발 방법론, 인공지능), XML, Grid 컴퓨팅



### 박 인 수

e-mail : ispark@ats.go.kr

1988년 원광대학교 전자공학과(학사)

1992년 원광대학교 전자계산기공학과(석사)

1978년~1992년 전북지방공산품검사소 연구원

1992년~1994년 전남지방공업기술원 기술 지원과장

1994년~현재 산업자원부 기술표준원 공업연구관

2000년~현재 한국산업기술대학교 컴퓨터공학과 겸임교수

2001년~현재 한국산업표준심의회 소프트웨어공학분야 전문위원 (ISO/IEC JTC1/SC7/WG6 국제표준 전문가 활동)

2002년~현재 호서대학교 벤처전문대학원 컴퓨터응용기술학과 박사과정

관심분야 : 소프트웨어공학(S/W품질평가 및 표준적합성 인증), 임베디드 S/W, 계측·제어, 프로그래밍