서비스지향구조 기반 소프트웨어의 유지보수성 시험 평가 방법

Maintainability Testing Evaluation Method for Service Oriented Architecture based Software

김진식, 전인오
호서대학교 벤처전문대학원 정보경영학과

Jin-Sik Kim(jiskim@dtaq.re.kr), In-Oh Jeon(eric@office.hoseo.ac.kr)

요약

서비스지향구조 소프트웨어 품질평가는 품질 향상을 유도할 수 있는 기술로서 국제 표준을 수용하는 전략기술 개발을 통해 객관성과 활용도를 높일 수 있고 많은 소프트웨어 기업들이 비즈니스 요구사항과 연계된 솔루션을 구현하기 위한 방법으로 서비스지향구조 기법을 사용하고 있다. 또한, 정부차원의 시범사업이 추진되고 있으며 관련 업계의 기술 개발에 따라 서비스지향구조 기반 소프트웨어의 상용화가 확산되고 있다. 따라서, 본 연구에서는 서비스지향구조 기반 소프트웨어의 유지보수성 품질을 평가하기 위해 시험 메트릭을 제안하였고 품질을 측정하고 그 결과를 적절한 기준에 따라 판정하는 방법에 대해 연구를 수행하고 평가 사례를 제시하여 평가 방법을 명확히 제안하였다. 본 연구를 통해 서비스지향구조 기반 소프트웨어의 유지보수성 품질향상을 유도하고 서비스지향구조 기반 소프트웨어에 대한 양적/질적인 수요를 충족할 수 있다고 본다.

■ 중심어 : | 서비스지향구조 | 유지보수성 | 시험평가 |

Abstract

Service oriented architecture software quality estimation is using Service oriented architecture techniques by method can improve objectivity and practical use degree through strategy technology development that accommodate international standard as technology that can motive quality sophistication and many software corporations embody business requirement and solution to associate. Also, setting an example business of government dimension is propeled and common use anger of Service oriented architecture base software according to technical development of related business circles is spreading. Therefore, proposed examination Metrik and measure quality and achieve research about method to decide the result according to suitable standard and present estimation example and present definitely estimation method to estimate Maintainability quality of Service oriented architecture base software in this research. Motive Maintainability quality sophistication of Service oriented architecture base software through this research and about Service oriented architecture base software quantitative / that can fulfill qualitative demand see.

■ keyword : | Service oriented architecture | Maintainability | Testing Evaluation |

접수번호 : #090211-004
접수일자 : 2009년 02월 11일
심사완료일 : 2009년 03월 05일
교신저자 : 김진식, e-mail : jiskim@dtaq.re.kr

I. 서 론

최근의 비즈니스 환경은 과거 독립적인 조직 및 프로세스에 의해 주도되는 수직적 통합에서 고객, 공급자, 파트너 등 다수 기업과의 관계적 협업 관계가 중시되는 수평적 통합 환경으로 변화하고 있다. 이러한 관계적 협업이 중시되는 비즈니스 환경에서 경쟁력 있는 기업은 급변하는 시장요구에 민첩하게 대응할 수 있어야 한다[7][8]. 이러한 비즈니스 요구에 효율적으로 대응하기 위한 기업 IT 아키텍처로 대표되는 것이 '서비스지향구조(Service Oriented Architecture)'이다. 서비스지향구조는 전통적인 프로그램 중심의 설계/개발 방식에서 비즈니스 프로세스 관점에서 재활용 가능한 단위로 서비스를 설계/개발하게 함으로써 특정 프로세스나 서비스 변경 또는 내부/외부 시스템과의 비즈니스 통합시 효율적이고 빠른 대응이 가능하다는 점에서 그 의미가 깊다[9]. 서비스지향구조는 특정 기술이나 플랫폼에 종속되지 않고 느슨한 결합(Loosely Coupled)을 가지고 상호 연동할 수 있는 서비스들의 조합으로 어플리케이션 개발을 가능하게 하는 정보시스템 아키텍처이다[10]. 서비스지향구조 기반 소프트웨어의 상용화가 급격히 진전되고 있는 시점에서 이에 따른 서비스지향구조 기반 소프트웨어의 품질평가 요구에 대응하기 위해, 본 연구에서는 서비스지향구조 기반 소프트웨어 분야의 기반기술을 조사/분석하고 서비스지향구조 기반 소프트웨어의 품질특성을 분석함으로써 서비스지향구조 기반 소프트웨어의 품질을 시험하여 측정하고 그 결과를 적절한 기준에 따라 판정하는 평가모델을 개발하고자 한다. 서비스지향구조 기반 소프트웨어의 품질 평가모델 개발에 관한 본 연구의 중요성을 기술적 측면에서 살펴보면 서비스지향구조 기반 소프트웨어의 품질 향상을 유도할 수 있는 기술로서 국제 표준을 수용하는 전략기술 개발을 통해 객관성과 활용도를 높일 수 있으며 서비스지향구조 기반 소프트웨어 사용자의 요구를 개발자가 적절히 수용할 수 있도록 하고 품질 향상 및 신뢰성 높은 소프트웨어의 개발을 유도할 수 있는 기술로서, 선진국의 기술 보호 장벽에 대처할 수 있는 핵심 기술이라 할 수 있으며 서비스지향구조 기반 소프트웨어와 부합하도록 애플리케이션들을 통합하면 비용과 노력이 대폭 줄어들며 서비스 지향의 본질적인 재사용 촉진은 서비스 지향 솔루션 구축에 있어 비용과 노력을 감소시키는 장점이 있다.

따라서, 본 연구의 2장에서는 서비스지향구조 기반 소프트웨어의 동향과 품질시험 동향에 대해 기술하였으며 3장에서는 서비스지향구조 기반 소프트웨어의 장점과 요구사항을 4장과 5장에서는 각각 서비스지향구조 기반 소프트웨어의 유지보수성의 체계 및 시험모듈을 기술하였으며 또한, 6장에서는 서비스지향구조 기반 소프트웨어의 시험평가 방법과 특성별 시험 결과를 기술하였다. 끝으로, 7장에서는 성능 시험 사례와 그 결과를 기술하였다.

II. 관련 연구

1. 서비스지향구조 기반 소프트웨어의 동향

1990년대 말 이후 전자정부의 추진은 인터넷을 통해 기존의 행정 프로세스를 향상시키기 위한 수단으로 시작되었다. 1995년 이전 세계적으로 50여개에 불과하던 전자정부 관련 사이트가 2001년에는 50만개 이상으로 급격히 증가하였다. 이처럼 최근 들어 많은 국가들이 전자정보를 구축하기 위해 노력하고 있으며 특히 미국, 캐나다, 유럽의 여러 국가들은 가장 모범적인 구축 사례를 보여주고 있다.

미국 정부는 전자정부의 아키텍처를 개별 부처 및 기관별 서비스로부터 공통서비스로 전환하는 것을 목표로 하고 있다. 또한 OMB에서 배포한 EA 평가프레임워크 2.0을 통해 서비스지향구조를 IT개발 운영관리 구조로서 추진하고 있다.

덴마크는 정부차원에서 전자정부를 지원하기 위해 주요 정보 아키텍처에 XML을 접목시켰다. 또한 서비스지향구조를 이용해 조화된 서비스 개발과 재사용으로 기업의 역할을 강화하고 있다. 그리고, 이탈리아 정부는 비즈니스 커뮤니티를 만들기 위해 비즈니스 커뮤니티 서비스 기반구조(Business Community Service Infrastructure : BCSI)라 불리는 네트워크 서비스 구조

를 설계하는 프로젝트를 수행하였다. 기존의 네트워크 서비스와 새롭게 개발될 서비스의 통합을 원활히 하기 위해서 서비스지향구조를 기반으로 XML과 웹 서비스 기술을 도입하여 BCSI를 구축하고 있다. 현재 우리나라는 전자정부의 발전단계 모형 중 2단계 '온라인화' 단계를 마치고 3단계인 '통합'의 단계로 진행하고 있다. 우리를 포함한 선진국형 전자정부는 기관별 전산화 단계에서 이음새 없는 서비스 제공을 통한 업무혁신 단계로 진화하는데 전자정부 사업 예산을 투자하고 있으며, 많은 사업들이 부처 간의 협업과 연계에 초점을 맞추어 추진되고 있다.

2. 서비스지향구조 소프트웨어의 시험 필요성

서비스지향구조는 변화에 적시에 적절히 대응할 수 있는 IT 시스템을 구축할 수 있도록 하는 개념이다. 이를 위해 조직 내부 프로세스, 어플리케이션들을 각각 '서비스'라는 기본적인 기능단위로 나누고 이들 서비스를 연결하여 원하는 기능을 하도록 구성하였을 때 이 변화를 반영하여 서비스의 연결 구성을 변화시켜 새로운 기능을 제공하도록 쉽고 빠르게 구성할 수 있도록 하는 것이다.

서비스지향구조의 특징은 4가지로 나누어 볼 수 있는데 첫째, 유연성으로 서비스 컴포넌트 단위의 조립, 재조합이 간단하며 둘째, 재사용성으로 공통으로 이용할 수 있는 서비스 컴포넌트를 만들고 기존 시스템을 서비스 컴포넌트화하며 셋째, 확장성 및 통합성으로 조직 및 회사 전체에 걸쳐 시스템 어플리케이션 간의 데이터 연계가 용이하고 업무 프로세스 간의 제어를 자동화시키며 마지막으로 감시, 모니터링으로 프로세스 이벤트의 모니터링 및 이벤트 취득을 위한 표준을 기반 한다. 이와 같은 서비스지향구조의 특징은 서비스지향구조 기반 소프트웨어를 통해 반영되며 일반적인 소프트웨어가 갖는 특성에 부가하여 서비스지향구조 기반 소프트웨어만이 효과적으로 지원할 수 있는 특성이라고 할 수 있다. 따라서 서비스지향구조 기반 소프트웨어는 일반적인 소프트웨어가 갖는 ISO/IEC 9126 등의 국제품질 표준을 적용하여 품질특성과 더불어 서비스지향구조 기반 소프트웨어만이 갖는 특성을 포함한 품질특성을 기반으로 평가모델을 개발하여 적용해야 할 필요가 있다[6].

Ⅲ. 서비스지향구조의 장점과 요구사항

1. 통합 비용 절감

시스템 간의 통합은 일회성(one-off), 포인트-투-포인트(point-to-point) 접속 방식으로 수행된다. 이 솔루션은 한 시스템의 특정 데이터 세트를 다른 시스템으로 이동할 때 발생하는 문제를 해결한다. 이는 전용 API와 재사용할 수 없는 특수 코드로 작성된 데이터를 사용한다. 이와 같은 사항은 일정한 수준의 복잡성을 안고 있을 뿐 아니라 사전에 전사적인 정보 통합을 요구하기 때문에 여러 가지 문제를 발생시키게 된다.

예를 들어, 기업이 고객 데이터에 액세스해야 하는 영업 포털을 구현하고 있다고 가정해 보면, 포털은 영업 담당자들에게 고객 정보를 제공하기 위해 CICS 애플리케이션에 액세스해야 하는데 위에서 설명한 통합 시스템은 그다지 도움이 되지 않는다. 즉, 통합해야 하는 N 개의 시스템이 있을 경우 연결이 필요한 총수는 N/2*(N-1)이다. 이들 각 연결은 다른 연결에서 거의 재사용하지 않거나 아예 없는 포인트 솔루션이 될 수가 있다. 대신 각 시스템에 대한 표준 기반 서비스 인터페이스를 구축하고 여기에 모든 시스템이 연결되도록 한다면, 각 시스템에 이러한 통합 작업을 단 한 번만 수행하면 되기 때문에 시스템 수가 늘어나더라도 비용을 절감하게 된다.

서비스지향구조의 이러한 이점은 가장 분명하고 가장 쉽게 눈으로 확인할 수 있지만 서비스지향구조 를 구현하는 가장 중요한 이유는 아니다. 실제로 강력히 결합된 모놀리식 엔터프라이즈 정보시스템의 복잡성을 관리하는 문제가 구현 비용 문제보다 더욱 심각한 사례가 자주 발생하고 있다.

2. 투명성과 자율성

많은 수의 IT 시스템이 서로 다른 기술을 사용하면서 연결되어 있다면 종속성 문제로 인해 시스템 일부를 변

경하는 것이 매우 어렵다. 이러한 부작용의 근본 원인을 찾는다는 것은 불가능하지 않지만 매우 어렵다고 볼 수 있다. 일반적으로 백로그가 길어지거나 비용이 증가하면 시스템을 변경할 수 없는 등의 결과가 초래된다. 변경의 영향을 완벽하게 분석할 수 있다 하더라도 변경 작업에 소요되는 비용이 그 이점을 무의미하게 만들 수 있다.

독립 시스템은 투명성이나 유연성이 없는 하나의 모놀리식 블록이 되었다. 이를 해결하는 유일한 방법은 시스템을 상호 연결하는 방식을 변경하는 것이다. 완벽하게 정의된 계약을 사용하여 다른 시스템과 상호 작용하면서 독자적으로 운영되도록 시스템을 서비스에 매핑해야 한다. 이는 중앙 집중식 시스템 및 자율 시스템 측면 모두에서 이익을 실현할 수 있다. 전체 인프라를 이러한 방식으로 세분화하면 시스템 간의 종속성을 훨씬 쉽게 추적할 수 있다. 가입자 정보를 이용해 서비스 레지스트리에 표준 기반 서비스 인터페이스의 카탈로그를 만들어 전체 인프라의 종속성 관리를 수행할 수 있다. 또 다른 이점은 IT 그룹이 훨씬 자율적인 방식으로 다른 그룹에서 제공하는 서비스를 사용할 수 있다는 것이다. 서비스는 표준화된 계약을 통해 중앙 레지스트리에서 제공되기 때문에 손쉽게 이들을 검색해 상호 작용할 수 있다. 결국 " 수동적인" 조정 없이 재사용이 증가함에 따라 서비스 가치가 높아지게 된다.

3. 서비스 수준의 업계 표준 준수

서비스지향구조의 또 다른 장점은 업계 표준의 준수로써 이 장점은 두 가지 범주로 나눌 수 있다.

첫째, IT 인프라가 표준 기반 계약을 사용하여 상호 작용하는 독립적이고 자율적인 서비스의 집합체일 경우 다른 비즈니스와의 작업을 통합하는 것이 훨씬 쉬워진다. 사실, 점차 많은 서비스가 보급되면서 내부 서비스를 더 저렴하게 제공하는 비즈니스 파트너의 서비스로 교체하거나 내부 서비스를 시장에서 경쟁력 있는 서비스로 전환할 때 이 IT 변환이 새로운 비즈니스 기회를 창출할 수 있다.

둘째, 업계 표준은 다른 측면에서도 이점을 제공한다. 모든 유형의 서비스를 구축하기 위해서는 툴과 프레임워크가 필요하다. 대부분의 업종이 XML 과 웹 서비스를 기반으로 하는 서비스지향구조 의 공통된 단일 접근 방식으로 통합되고 있기 때문에 서비스 네트워크를 구축하고 운영하는 데 필요한 툴과 프레임워크가 더 일반화되고 널리 사용할 수 있게 되었다.

4. 이기종 환경에서 잘 동작

서비스지향구조는 대부분의 기업에서 발견되는 '모든 것을 갖고 있는' IT 환경을 통합하는 일을 좀 더 쉽게 해 준다. 즉, 서비스지향구조가 제시하는 가장 큰 가치 가운데 하나는 이기종 환경에 적합한 구조라고 할 수 있다. 서비스지향구조를 통하면, 개발자들은 애플리케이션들을 연결하는 새로운 코드를 작성하기 위해 과도한 시간을 낭비할 필요가 없어진다. 대신 개발자들은 웹서비스 같은 표준 프로토콜을 사용할 수 있으며, 서비스지향구조 코드의 상당 부분은 재사용이 가능하기 때문에 개발 비용도 줄어든다.

5. 기존 포트폴리오의 활용

서비스지향구조의 장점은 기존의 포트폴리오를 활용할 수 있다는 점으로 기존 시스템을 제거하고 새로운 시스템으로 대체할 필요가 없어진다. 기존 시스템의 기능들을 파악한 후 그것들을 활용함으로서, CIO는 위험을 최소화하면서 기존 IT 투자의 가치를 극대화할 수 있다.

또 서비스(예를 들면, 단순객체 접근프로토콜(서비스지향구조P: simple object access protocol)과 웹

서비스 기술언어(WSDL: Web services description language) 등을 사용하는 서비스)를 구축하면, 내부 프로세스가 심플해지고 고객과 비즈니스 파트너들과 더 쉽게 정보를 공유할 수 있게 된다.

Ⅳ. 서비스지향구조의 유지보수성의 체계

서비스지향구조에서의 유지보수성은 소프트웨어 제품이 변경되는 능력, 변경에는 환경과 요구사항 및 기능적 명세에 따른 소프트웨어의 수정, 개선, 혹은 개작

등이 포함된다.

국제 품질 표준에서 유지보수성은 품질 주특성 중에서 중요한 위치를 차지하고 있으며 서비스지향구조 소프트웨어의 장점과 요구사항을 평가할 수 있는 특성이라고 할 수 있다. 유지보수성을 측정하기위해서는 품질 부특성 5가지 즉, 분석성, 변경성, 안정성, 시험성, 준수성을 정량적으로 시험하여 평가하여야 한다[1][2]. 유지보수성의 품질 부특성은 [그림 1]과 같이 나타내었다.

- 분석성(Analyzability) : 소프트웨어의 결함이나 고장의 원인 혹은 변경될 부분들의 식별에 대한 진단을 가능하게 하는 소프트웨어 제품의 능력이다.
- 변경성(Changeability) : 명세된 변경이 구현될 수 있도록 하는 소프트웨어 제품의 능력으로 여기에서 구현은 코딩, 설계, 문서화 등의 변경을 포함하며 소프트웨어가 최종 사용자에 의해 변경된다면 변경성은 운용성에 영향을 미칠 수 있다.
- 안정성(Stability) : 소프트웨어 변경으로 인한 예상치 않은 결과를 최소화하는 소프트웨어 능력이다.
- 시험성(Testability) : 변경된 소프트웨어가 확인될 수 있는 소프트웨어 제품의 능력이다.
- 준수성(Compliance) : 유지보수성과 관련된 표준 및 관례를 고수하는 소프트웨어 능력이다.

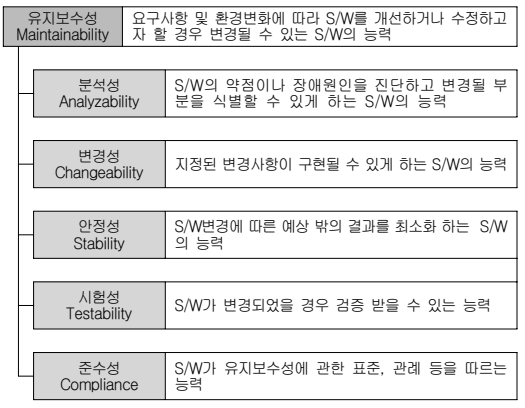

그림 1. 유지보수성의 품질 부특성

V. 서비스지향구조 기반 소프트웨어의 유지보수성 시험 모듈

서비스지향구조 기반 소프트웨어의 유지보수성은 규정된 수정을 수행하기 위하여 필요한 노력과 관련된 속성의 집합으로 수정이나 오류 교정, 환경

변화 및 요구사항과 기능 명세서 변경에 따른 소프트웨어 개선이나 적응을 말한다.

시험 모듈은 품질시험 과정에서 활용할 수 있는 자료로서 품질시험원에게 필요한 최소 필요사항을 포함하여 [표 1]과 같이 테이블의 형태로 구성하였다. 시험 과정에서 필요한 세부 사항은 품질시험 모듈을 참조할 수 있다.

VI. 서비스지향구조 소프트웨어 시험 평가

이 절에서는 서비스지향구조 소프트웨어를 시험하기 위해 필요한 제반 환경에 대해 기술하였다. 먼저, 서비스지향구조 소프트웨어 시험을 위한 기본적인 환경에 대해 살펴보기로 한다.

1. 시험절차

서비스지향구조 소프트웨어의 시험 평가를 위해서는 먼저 제품에 대한 정확한 이해가 필요하며 제품을 정확히 분석하여 테스트케이스를 구성하여야 한다.

본 연구에서 제안한 서비스지향구조 소프트웨어의 품질평가 체계와 메트릭은 객관적인 시험을 통해 시험 대상 제품이 소프트웨어 품질 요구사항을 만족하는지 확인하고, 소프트웨어 품질을 개선함으로써 제품의 신뢰도 향상 및 경쟁력을 향상 시킬 수 있다.

본 연구에서 제안한 서비스지향구조 소프트웨어 의 시험 평가를 위해서는 다음과 같은 방법을 취할 수 있다. 먼저 시험을 하고자 하는 제품의 시험 계획을 수립하고 다음은 시험 환경을 구축하여 시험 할 수 있도록 하고 제안한 메트릭에 대하여 시험 항목을 도출하며 테스트 시나리오와 케이스를 작성하고 시험을 수행한다. 시험 도중 발생하는 결함에 대해서는 결함리포트를 작성하고 결함리스트를 중심으로 하여 제품의 패치를 통해서 결함을 수정하고 다시 제품에 대한 회귀시험을 수행하여 시험결과를 작성하는 순서로 진행한다.

표 1. 유지보수성 측정을 위한 시험 모듈의 예

부특성	메트릭명	측정항목		계산식	결과영역	적용 대상
분석성	진단기능 정보제공	A	진단기능에 관한 정보 제공 여부	진단기능 정보 제공 (DAP) = A	0 ≤ 권리 표현 지원 ≤ 1	공통
	소프트웨어를 사용할 때, 발생하는 오류의 증상 및 해결할 수 있는 진단기능에 관한 정보를 제공하고 있습니까?		제품설명서 또는 사용자 문서에 소프트웨어를 사용할 때 발생하는 오류의 증상과 원인을 알아내고 해결할 수 있는 진단기능에 대한 정보 제공 여부			
변경성	환경설정 변경 정보제공	A	환경 설정에 대한 정보 제공 여부	환경설정 변경 정보제공 (CIP) = A	환경설정 변경 정보제공 (CIP) = Y or N or NA	
	사용자 설치가 가능한 경우, 소프트웨어 제품을 시스템에 설치(설치 재시도)하기 위한 정보와 설치 프로그램을 제공하고 있습니까?		(환경 설정에 대한 정보) 제품설명서와 사용자 문서에 소프트웨어의 환경 및 기능이 변경될 수 있는지에 대한 정보를 기술 변경 가능한 항목의 설명 및 방법, 그 결과가 미치는 영향 등 기술			
안정성	환경 설정 변경 안정성 정보 제공	A	환경 변경으로 인해 발생할 가능성이 있는 오류에 대한 정보 제공 여부	환경설정 변경 안정성 정보제공 (SIP) = A	환경설정 변경 안정성 정보제공 (SIP) = Y or N or NA	
	소프트웨어의 환경 설정 변경으로 인해 발생할 수 있는 예상치 못한 결과에 대한 정보를 제공하고 있습니까?		제품설명서와 사용자 문서에 소프트웨어의 환경 설정 변경으로 인해 발생할 수 있는 예상치 못한 결과에 대한 정보가 기술되어 있는지 평가			
시험 가능성	내장형 시험 기능 정보 제공	A	내장형 시험기능 정보제공 여부	내장형 시험기능 정보제공 (BTP) = A	내장형 시험기능 정보제공 (BTP) = Y or N or NA	
	소프트웨어가 스스로 시험할 수 있는 내장형 시험 기능에 대한 정보를 제공하고 있습니까?		(내장형 시험 기능의 예) 시뮬레이션 기능, 사전-체크 기능 등			
준수성	유지보수 표준 준수 정보 제공	A	유지보수 관련 표준 정보 제공 여부	유지보수 표준 준수 정보제공 (MSP) = A	유지보수 표준 준수 정보제공 (MSP) = Y or N or NA	
	유지 보수와 관련된 표준, 규약 등의 정보를 제공하고 있습니까?		(유지보수 관련 표준 정보의 예) 제품설명서, 사용자 문서에 기술되어 있는 유지보수 관련 표준, 규약, 협약 등에 대한 정보			

2. 시험환경 구축

시험대상 소프트웨어는 서비스지향구조 기반으로 개발된 웹기반 메일 송수신, 웹오피스, 웹폴더 등의 기능을 제공하는 웹메일 솔루션으로써 주요기능으로는 웹기반 메일 송수신, 웹오피스, 웹폴더, 이레터, 주소록, 쪽지함, 환경설정, 통합관리등이 있다.

평가대상 제품의 소프트웨어를 시험하기 위해 [1]번 메일 서버에 설치한 프로그램으로는 시험 대상

제품 [2]~[4]번 클라이언트에 설치한 프로그램으로는 Internet Explorer 6.0과 기타 응용 프로그램 : MS Office 2007, 한글 2005, PDF Viewer 등이 사용되었으며 네트워크는 10/100Mbps 스위칭 허브를 사용하였다. 성능 측정 도구로는 [1]번 메일 서버에 Team Quest Manager v10.1을 설치하였고 [2]번 클라이언트에는 Team Quest Viewer v10.1와 Borland Silk Performer 2007(부하 발생 도구)을 설치하고 [5]번 부하발생기에 XP를 설치하였다.

3. 유지보수성과 일반적 요구사항의 품질특성

소프트웨어 품질 평가에 있어서 유지보수성은 소프트웨어 제품이 변경되는 능력을 의미하며 변경에는 환경과 요구사항 및 기능적 명세에 따른 소프트웨어의 수정, 개선, 또는 개작 등이 포함된 능력을 평가하는 것이며 부특성으로는 분석성, 변경성, 안정성, 시험가능성, 준수성등이 있고 일반적 요구사항으로는 제품 정보 제공 능력 및 제품의 바이러스 감염 여부의 속성집합을 도출할 수 있으며 이를 정리하면 [표 2]와 같다.

표 2. 유지보수성 및 일반적 요구사항의 품질 특성

유지보수성	분석성	발생하는 결함의 증상 및 장애원인을 진단하고 변경될 부분을 식별 할 수 있게 하는 능력
	변경성	환경 설정이 변경 될 수 있게 하는 능력
	안정성	환경 설정 변경에 따른 예상 밖의 결과에 대한 정보를 제공하는 능력
	시험가능성	소프트웨어가 스스로 시험할 수 있는 능력(내장형 시험기능)
	준수성	유지보수성에 관한 표준, 관례 등을 따르는 능력
일반적 요구사항	식별 및 표시	제품정보 제공 능력
	안전성	바이러스 감염여부 능력

4. 품질 특성별 시험 결과

시험대상 제품은 유지보수성과 일반적 요구사항을 중심으로 시험하였으며 제안된 메트릭을 중심으로 6가지 품질 특성에 대한 시험 결과를 기술하였으며 예를 들어, [표 3]과 같이 각 특성별 최종 결함수의 집계표를 작성하였다.

표 3. 시험 전/후의 결함 수

품질특성	수정전 결함수	최종 결함수
유지보수성	2	0
일반적 요구사항	4	0
계	6	0

4.1 유지보수성 및 일반적 요구사항의 시험 결과

유지보수성을 시험한 결과 관리자 권한 설정 오류와 메일용량 설정 오류 등이 발생하였으나 수정 보완 및 회귀시험 과정을 거친 후 최종적으로 이상이 없음을 확인하였고 일반적 요구사항의 결함을 테스트한 결과로는 제품 구동 환경 정보 미제공과 제품 식별 정보 미제공, 그리고 제품 공급자 정보 미제공 등의 결함이 발생하였으나, 수정 보완 및 회귀시험 과정을 거친 후 최종적으로 정확한 정보가 제공됨을 확인하였으며 [표 4]와 같이 정리하였다.

4.2 결함정도별 분류

품질 특성별 작성된 결과표를 중심으로 각 결함에 대해서 결함의 정도를 파악한다. 결함의 정도는 각각의 기준에 따라서 설정하며 시험결과 High가 11개, Medium이 15개, Low가 8개 발견 되었다.

표 4. 유지보수성 및 일반적 요구사항의 결함 속성

구 분	결함수	결함 정도	결함요약	최종 결함수
유지 보수성	2	H	관리자 권한 부여 오류	0
		M	메일 용량 초과 오류	
일반적 요구사항	4	M	인도 항목 정보 미제공	0
		M	제품 공급자 정보 미제공	
		M	제품 구동 환경 정보 미제공	
		M	제품 식별 정보 미제공	

결함 정도의 분류는 H(High)인 경우 기능이 정상적으로 동작하지 않거나, 하드웨어 시스템 혹은 프로그램이 비정상적으로 종료되는 등의 치명적인 결함이 발생하는 경우로 분류하며, M(Medium)은 프로그램 운영에는 문제가 없으나, 기능이 정확하게 동작하지 않거나 사용자의 혼란을 야기하는 정도의 결함이 발생하는 경우로 분류할 수 있으며, L(Low)은 프로그램 운영에 문제가 없고, 기능도 정확하게 동작하나 권고 사항 수준의 경미한 결함이 발생하는 경우를 들 수 있다. 그리고 제품설명서를 중심으로 하여 일반적 요구사항에 대한 평가를 실시한다. 예를 들어, 프로그램에 버전이 명시되어 있지 않는 결함이 발생하거나 기타 제품 설명서에 제품에 대한 기능상의 문제가 없는지를 확인하면서 평가한다.

Ⅷ. 성능시험 및 결과

성능시험 시나리오는 [표 5]와 같이 시나리오 1, 2로 나누어서 시나리오 1에서는 다수의 동시 사용자가 웹메일 사이트에 로그인 후 5초 뒤 로그아웃하여 측정을 하였으며 시나리오 2에서는 다수의 동시사용자가 웹메일 서버를 이용하여 본문이 100KB인 메일을 송수신하여 측정하였다.

표 5. 성능시험 시나리오

시나리오 번호	설 명
시나리오-1	다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 사이트에 로그인하고 5초 후 로그아웃
시나리오-2	다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 서버를 이용하여 본문이 100KB인 메일 송수신

성능시험 결과는 [표 6]과 같이 자원효율성과 시간효율성을 측정하였으며, 자원효율성의 시험결과는 CPU 사용률과 메모리 사용률을 측정하였으며 시간효율성은 다수의 동시 사용자가 로그인후 5초뒤 로그아웃할 경우 응답시간을 측정하였다.

표 6. 측정항목

항목	내용	단위
CPU 사용률	usr : 사용자 프로세스가 CPU를 사용하는 시간의 백분율	%
	sys : 시스템 프로세스가 CPU를 사용하는 시간의 백분율	
	idle : CPU를 사용하지 않은 시간의 백분율	
	wio : 입출력 대기 시간의 백분율	
메모리 사용량	freemem : 사용하지 않는 물리 메모리의 양	MB
	buffermem : 버퍼에서 사용된 메모리의 양	
	cachedmem : 페이지 캐시에서 사용된 메모리의 양	
응답 시간	시스템에 조회나 요구 등을 입력한 직후부터 해당 명령의 처리가 완료된 시점까지 소요된 시간	초

1. 자원효율성

[그림 2]는 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 사이트에 로그인하고 5초 후 로그아웃 했을 때의 CPU사용량을 그래프로 나타낸 것이다. 사용자의 수가 증가함에 따라 CPU의 사용량이 점차적으로 증가함을 알 수 있다.

[그림 3]은 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 사이트에 로그인하고 5초 후 로그아웃 했을 때의 메모리 사용량을 나타낸 것으로 사용자의 수가 증가함에도 크게 변동이 없이 일정함을 알 수 있다.

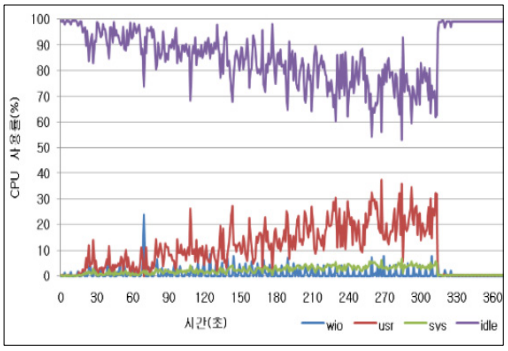

그림 2. 메일 서버 - CPU 사용률

[그림 4]는 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 서버를 이용하여 본문이

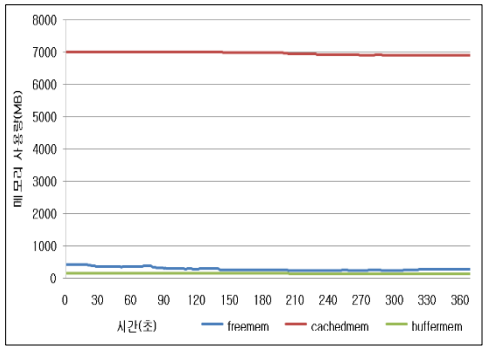

그림 3. 메일 서버 - 메모리 사용량

100KB인 메일 송수신을 했을때 CPU사용량을 나타낸 것으로 유저가 메일 송수신을 시작 했을 때부터 CPU사용량과 입출력 대기시간이 크게 증가함을 알 수 있다.

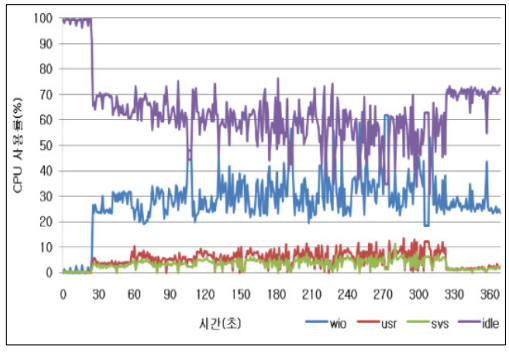

그림 4. 메일 서버 - CPU 사용률

[그림 5]는 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 서버를 이용하여 본문이 100KB인 메일 송수신을 했을 때 메모리 사용량을 측정한 것으로 점차적으로 사용자와 시간이 증가함에 따라 페이지 캐시에서 사용된 메모리의 양이 7000MB에서 8000MB로 증가하였고 사용하지 않는 물리 메모리의 양은 약 500MB에서 0가까이로 줄었음을 알 수 있다.

즉, 자원효율성은 CPU사용률과 메모리 사용량을 측정하였으며 다수의 동시 사용자(최초 20명, 5분동안 30초당 20명씩 증가)가 웹메일 서버를 이용하여 본문이 100KB인 메일을 송수신할 경우, CPU 사용률은 최고 65%까지 증가하고 큐에 쌓여있는 메일이 모두 송수신

될 때까지 약 27%의 CPU를 지속적으로 사용하나 이후 1%미만으로 감소하였다. 또한, 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 사이트에 로그인하고 5초 후 로그아웃할 경우, 메모리 사용량은 평균 7,873MB이며, 이후 안정적으로 유지되었고, 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 서버를 이용하여 본문이 100KB인 메일을 송수신할 경우, 메모리 사용량은 평균 7,935MB이며, 이후 안정적으로 유지되었다.

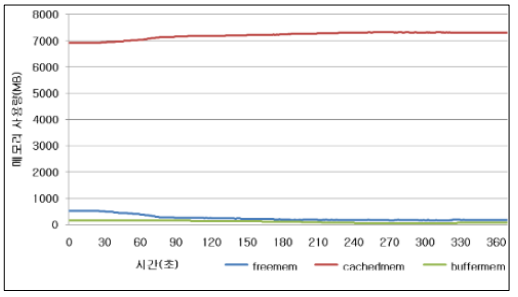

그림 5. 메일 서버 - 메모리 사용량

2. 시간효율성

[그림 6]은 다수의 동시 사용자(최초 20명, 5분 동안 30초당 20명씩 증가)가 웹메일 사이트에 로그인하고 5초 후 로그아웃 했을 때 시스템에 조회나 요구를 입력한 직후부터 해당 명령의 처리가 완료된 시점까지 소요된 시간을 측정한 값을 나타낸 것이다. 결과적으로 다수의 동시 사용자가 웹메일 사이트에 로그인하고 5초 후 로그아웃할 경우, 응답시간이 최소 1.47초, 최대 22.97, 평균 3.14초 소요되었다.

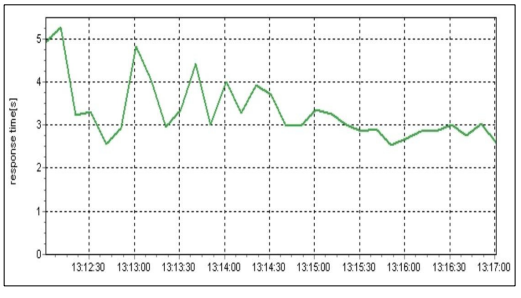

그림 6. 메일 서버 - 응답시간

Ⅷ. 결 론

컴퓨터 기술의 급격한 발전으로 오늘날 컴퓨터를 활용하지 않는 업무 분야가 거의 없을 정도로 많은 분야에서 컴퓨터의 활용도는 점점 높아지고 있을 뿐만 아니라 일반 개인 사용자들에게도 컴퓨터가 급속히 보급되고 있는 추세이다. 이러한 추세에 맞추어 컴퓨터를 어떤 용도로 사용하느냐에 따라 용도에 맞는 적절한 소프트웨어의 개발이 요구되고 있으며 이를 뒷받침 할 수 있도록 소프트웨어 개발 업체에서는 다양한 종류의 소프트웨어들을 개발하고 있다. 이제 사용자는 다양한 유형의 소프트웨어들 중에서 자신이 컴퓨터를 사용하는 목적과 용도에 알맞은 소프트웨어를 선택할 수 있게 되었으며 이로 인해 올바른 선택 방법에 대한 중요성이 대두되고 있다. 더불어 소프트웨어 제품의 품질이 중요한 관건으로 대두된 지 오래이며 소프트웨어 제품 품질에 대한 인증의 중요성이 높아짐에 따라 다양한 소프트웨어 유형에 따른 품질시험 및 인증 방법에 대한 연구가 추진되고 있다. 본 연구에서는 ISO/IEC 9126을 기반으로 서비스지향구조 소프트웨어의 유지보수성의 품질평가 모델을 개발하고 평가 과정에서 활용할 수 있는 품질검사표를 개발하였으며 시나리오를 작성해 성능시험을 테스트한 결과를 통하여 자원효율성과 시간효율성을 측정하는 방안을 모색하였다. 또한, 서비스지향구조 소프트웨어의 유지보수성에 대한 품질평가 방법을 구축하여 품질 수준의 제고를 위한 기준을 제시하였다. 서비스지향구조 기반 소프트웨어의 유지보수성에 대한 품질평가 모델 개발과 향후 실질적인 활용을 통해 서비스지향 구조 기반 소프트웨어의 품질수준을 향상시킬 수 있다고 본다.

참 고 문 헌

[1] ISO/IEC 9126, "Infomation Technology - Software Quality Characteris- tics and metrics - Part 1, 2, 3.

[2] ISO/IEC 14598, "Infomation Technology -

Software product evaluation - Part 1, 2, 3, 4, 5, 6.

[3] M. L. Cook, "Software Metrics : An Introduction and Annoted Bibliography," ACM SIGSOFT Software Engineering Notes, pp.41-60, Vol.7, No.2, 1982(4).

[4] W. M. Evangelist, "Software Complexity Metrics Sensitivity to Program Structuring Rules," The Journal of Systems and Software, 3, pp.231-243, Elsevier Science, 1983.

[5] M. Azuma, "Software Quality Evaluation System:Quality Models, Metrics and Processes - International Standards and Japanese Practice," Information and Software Technology, 1996.

[6] Thomas Erl, *Service-oriented Architecture : concept, technology, and design*, prentice Hall, 2005.

[7] R. Rich, reuse engineering for SOA, http://www.123.ibm.com/developerworks/ webservices/ library/ ws-rense/ soa.html, 2004.

[8] C. W. Kreuge, "software Reuse," ACM Computing Survey, pp.131-183, 1992.

[9] 권수갑, "SOA 개념과 동향", 전자부품연구원, 2005(11).

[10] 더크 크래프지그, 칼 방케, 더크 슬라마, "엔터프라이즈 서비스지향구조(서비스 지향 아키텍처 베스트 프랙티스)", 태극미디어, 2006(11).

[11] 토마스 얼, "서비스지향구조 : 서비스 지향 아키텍처(XML과 웹서비스 통합을 위한 필드 가이드)", 성안당, 2007(1).

[12] 양해술, "소프트웨어 시험평가 모듈 개선 연구", ETRI 컴퓨터 · 소프트웨어 기술연구소 위탁과제, 최종보고서, 2001(11).

저 자 소 개

김 진 식(Jin-Sik Kim) 정회원

- 1972년 : 인하대학교 전기공학과 졸업(학사)
- 1977년 : 인하대학교 대학원 토목공학과 졸업(공학석사)
- 1992년 : 고려대학교 대학원 재무관리전공(경영학석사)
- 2006년 : 호서대학교 벤처전문대학원 정보경영학과 박사과정
- 1978년 : 국방과학연구소(ADD) 통신전자실장
- 1985년 : 국방기술품질원 기동장비실장
- 2002년 : 국방벤처센터 센터장
- 2009년 ~ 현재 : 국방기술품질원 책임연구원

<관심분야> : 국방 벤처사업, 국방 소프트웨어 품질평가 및 감리, 국방 프로젝트 관리 및 평가

전 인 오(In-Oh Jeon) 정회원

- 1998년 : 호서대학교 전자공학과 졸업(학사)
- 2000년 : 중앙대학교 경영학과 졸업(석사)
- 2005년 : 호서대학교 소프트웨어 공학전공 졸업(공학박사)
- 1998년 ~ 2004년 : (주)씨아이정보기술 대표이사
- 2005년 ~ 현재 : 호서대학교 글로벌창업대학원 교수
- 2005년 ~ 현재 : 호서대학교 벤처전문대학원 교수

<관심분야> : 벤처창업론 및 컨설팅, 소프트웨어공학(특히, 소프트웨어 품질보증과 평가 및 품질감리), 전시/컨벤션산업