분산 소프트웨어 개발환경에 대한 확률 미분 방정식 모델을 이용한 최적 배포 문제

정회원 이 재 기*, 종신회원 남 상 식**

Optimal Release Problems based on a Stochastic Differential Equation Model Under the Distributed Software Development Environments

Jae-ki Lee* ***Regular Member***, Sang-sik Nam** ***Lifelong Member***

요 약

최근 소프트웨어 개발은 client/server 시스템이나 웹 프로그래밍, 객체지향 개발, 네트워크 환경에 의한 분산개발 등 새로운 개발 형태로써 다양하게 적용되고 있다. 한편 소프트웨어 분산 개발에 대한 기술도 관심이 되고 있으며, 객체지향 개념이 확대되고 있다. 이러한 기술에 의한 개발 작업량의 대폭 삭감이나 소프트웨어 품질 및 생산 개선의 효과가 점차 증대되어 가는 추세로 향후 광범위한 분야에 분산된 다수의 워크스테이션에 의해 병행되어 개발된 객체(objects)를 이용한 분산개발의 발전에 대해 고찰한다. 본 논문에서는 이러한 분산 소프트웨어 개발환경을 대상으로 확률미분방정식 모델에 의한 소프트웨어 최적 배포문제를 논한다. 과거에는 소프트웨어 개발 프로세스에 의한 출하 품질의 파악이나 시험 진도관리에 의한 신뢰성 평가를 행하는 접근방법(approach)에 의해 소프트웨어의 고장 발생 현상을 불확정 사상에 의해 확률, 통계적으로 취급하는 방법을 적용하였으나 본고에서는 fault 발견과정에서 계수에 의해 취급되는 비동차포아송과정(NHPP: Non-Homogeneous Poisson Process)에 의한 SRGM과 fault 발견 과정을 연속적으로 변동하는 확률 과정의 모델화된 확률 미분방정식(SDE: stochastic differential equation)에 의한 SRGM을 제안하여 최적의 배포시기를 결정한다. 여기서 시험단계 및 운용단계에 발생하는 비용 요인으로부터 도출된 총 소프트웨어 비용을 최소로 하는 시험시간인 최적 배포시기를 구한다. 특히, 총 소프트웨어 비용의 확률분포를 고려하여 최적 배포시기의 신뢰 한계도 논한다.

Key Words : 소프트웨어 신뢰성 평가, 분산개발환경, 확률미분방정식(SDE), 최적배포문제, 비용의 확률분포

ABSTRACT

Recently, Software Development was applied to new-approach methods as a various form : client-server system and web-programing, object-orient concept, distributed development with a network environments. On the other hand, it be concerned about the distributed development technology and increasing of object-oriented methodology. These technology is spread out the software quality and improve of software production, reduction of the software develop working. Futures, we considered about the distributed software development technique with a many workstation. In this paper, we discussed optimal release problems based on a stochastic differential equation model for a distributed Software development environments. In the past, the software reliability applied to quality a rough guess with a software development process and approach by the estimation of reliability for a test progress. But, in this paper, we decided to optimal release times two method: first, SRGM with an error counting model in fault detection phase by NHPP. Second, fault detection is change of continuous random variable by SDE(stochastic differential equation). Here, we decide to optimal release time as a minimum cost form the detected failure data and debugging fault data during the system test phase and operational phase. Especially, we discussed to limitation of reliability considering of total software cost probability distribution.

* 한국전자통신연구원 BcN시험기술팀 (jkilee@etri.re.kr), ** 한국전자통신연구원 BcN시험기술팀 (ssnam@etri.re.kr)
논문번호 : KICS2006-02-059, 접수일자 : 2006년 2월 2일, 최종논문접수일자 : 2006년 7월 10일

Ⅰ. 서 론

고도 정보화 사회에서의 컴퓨터시스템은 다양한 분야에 적용되고 있으며, 사회의 중심에서 중요한 역할을 담당하고 있다. 다시 말해서 컴퓨터시스템에 대한 하드웨어 분야는 기술혁신의 진전이나 비용에 대한 성능이 비약적으로 향상되었으며, 고 신뢰성 및 고기능성이 실현된 현시점에서는 소프트웨어 신뢰성의 저하가 문제시되고 있다. 또한 소프트웨어 수요가 급속히 증가되고 있고 또 복잡화, 다양화, 대규모화가 되는 경향이 있는바 이러한 개발에 의연하게 인위적인 작업을 행하는데 있어서 개발 중에 많은 소프트웨어 fault(또는 failure) 라고 불리는 결함(defect)이나 오류(error)가 잠입하게 된다. 이러한 fault에 의해 발생되는 소프트웨어 고장이 표면화를 통해 사용자에게 많은 영향을 주고 경우에 따라서는 인명에 다양한 형태의 중대 사고를 일으키는 경우도 있다. 이러한 현상을 줄이는 방법은 소프트웨어의 대표적 품질 특성인 신뢰성을 높이는 소프트웨어 개발을 통하는 것이 매우 중요하다. 컴퓨터 시스템의 고 신뢰화를 도식화한 문제에 대해서 중요한 구성요소의 하나인 소프트웨어의 고 신뢰화 실현기술로 소프트웨어 신뢰성을 정량적으로 평가하는 것은 개발단계를 과학적으로 관리하는 측면에서 특히 중요하다.[1]

소프트웨어 개발을 위한 환경은 client/server 시스템이나 웹 프로그래밍, 객체지향 개발, 네트워크 환경에 의한 분산 개발 등 새로운 개발 형태가 다양하게 사용되고 있다.[2-4] 특히, 최근의 급속한 인터넷의 보급과 소프트웨어 분산개발에 의한 기술이 주목받기 시작하였고 객체개념이 확대되고 있다. 이 기술에 의해 개발 작업량의 대폭 감소나 소프트웨어 품질 및 생산성 개선의 극대화로부터 향후 광범위하게 분산된 다수의 워크스테이션에 의해 병행 개발된 객체를 이용한 분산개발의 발전이 기대되고 있다. 이러한 추세에 맞추어 소프트웨어 시스템에 대한 개발 방법도 소프트웨어 개발환경의 급속한 변화에 유연하게 대응할 수 있는 대규모 분산 소프트웨어 시스템의 구성법과 전체의 품질을 확보할 수 있는 커다란 문제를 고려하여야 한다. 또한 업무나 사회 시스템에 대한 소프트웨어의 사회적 영향도 등을 실현하는 소프트웨어 개발 작업량 및 개발 비용을 증대시키는 방법이 있다. 이에 반해서 품질, 생산성에 대한 요구는 엄격하여 이에 대한 대책은 네트워크가 상호 접속된 분산 환경에 의해 개발된 소프트웨어 시스템의 신뢰성 평가에 대한 개발기간의 단축 및 개발비용의 삭감이 중요한 현실적인 과제가 되었다.

현재 사용되고 있는 분산 개발환경에 대한 유효 테스트 기법은 확립되지 않았으며, 본 논문의 대상인 테스트 전체의 흐름은 각 컴포넌트의 유닛(unit) 시험은 프로그램 작성자가 시험마다 일관되게 책임을 지는 것을 시행하고 시스템 전체의 통합으로부터 운용확인까지 종합시험의 개시 및 기간은 개발된 소프트웨어 시스템의 납기나 시험 진도를 포함하여 시스템 개발관리자의 결정에 따른다.

본 논문에서는 이러한 분산 소프트웨어 개발환경하에서 개발되는 확률적 미분방정식 모델에 의한 소프트웨어 최적 배포문제를 다룬다. 다시 말해서 종합 시험단계 종료시 실제 운용단계에 이르기 위한 최적 시기를 총 소프트웨어 비용을 최소로 하는 시간을 구하는 문제를 논한다. 과거의 최적 배포문제는 소프트웨어 비용의 기대치를 최소로 확정하는 최적 배포시기를 구하였다. 본 논문에서는 소프트웨어 비용을 확률변수로 소프트웨어 개발관리상의 유익한 배포시기 판정지표에 대해 소프트웨어 비용을 최소로 하는 최적 배포시기의 신뢰구간을 도출하는데 있다. 이것에 의해 소프트웨어 비용의 신뢰구간에 기반을 둔 최적 배포시기 및 총 기대 소프트웨어 비용의 존재 범위를 확인하는 것이 가능하다.

Ⅱ. 분산개발환경에 대한 소프트웨어 신뢰성 평가

분산 개발환경 상태에 의한 소프트웨어 개발의 특징은 객체지향기술(OOT : object oriented technology)에 의한 소프트웨어 부품화가 시행되고 있다. 이러한 소프트웨어 부품화는 소프트웨어 자체도 표준화에 의한 재이용이 가능하고 또 이 기술에 의해 개발 작업량이 대폭 삭감되거나 소프트웨어 품질 및 생산성 개선의 효과가 극대화 되는 것을 기대할 수 있다.[5]

분산 개발환경의 대상이 되는 소프트웨어 시스템이 대규모인 경우나 시스템에 포함하고 있는 기술된 언어의 종류나 시스템 내부의 구성요소가 많은 경우, 각 소프트웨어 컴포넌트간의 상호작용이 한층 혼재화 되어 있으면 시스템 시험단계에서 발견된 fault의 발생여부 파악이 어렵다.[6]

과거에는 소프트웨어 개발 프로세스에 의한 출하 품질의 파악이나 시험 진도관리에 의한 신뢰성 평가를 행하는 접근방법(approach)에 의해 소프트웨어

의 고장 발생 현상을 불확정 사상에 의해 확률, 통계적으로 취급하는 방법이 적용되었다. 이러한 일종의 소프트웨어 신뢰도 성장모델(SRGM: Software Reliability Growth Model)이 존재하였다.[7, 21-22]

분산 개발환경을 대상으로 한 소프트웨어 신뢰도 성장모델에 대하여 아래에 표시한 분산 개발환경의 통합 시험단계에 의한 fault 발견과정을 계수과정에 의해 취급되는 비동차포아송과정(NHPP : Non-Homogeneous Poisson Process)에 의한 SRGM과 이 fault 발견과정을 연속적으로 변동하는 확률과정의 모델화된 확률미분 방정식(SDE : Stochastic Differential Equation)[8]에 의한 SRGM을 제안한다.

(1) 분산 개발환경에 의한 NHPP에 기초로 한 SRGM[9] - NHPP 모델

$$H_c(t) = a\left(\sum_{i=1}^{n} P_i(1-e^{-b_i t}) + \sum_{j=1}^{m} P_{n+j}\left\{1-(1+b_{n+j}t).e^{-b_{n+j}t}\right\}\right) \quad (1)$$

여기서 $H_c(t)$는 시험시간 t마다 발견, 수정된 총 fault 수의 기대치를 표시한다.

a는 시험 시작 전에 잠재하고 있는 총 기대 fault 수를 $b_l(l=1,2,3,.....n+m)$은 l번째 컴포넌트에 대한 잔존 fault 1개당 fault 발견율을 표시한다. 한편 파라미터 $P_l(l=1,2,3,.....n+m)$은 l번째 컴포넌트에 대한 중복, 다시 말해서 각 소프트웨어 컴포넌트의 규모, 시험역량, 시험 툴 등에 영향을 주는 시험에 관한 중요도를 표시한다.

과거 다수의 소프트웨어 신뢰도 성장모델이 제안되었으며, 이 모델들은 실제 소프트웨어 개발현장에 개발지원 툴의 일부로써 실장 되었다. 그중에서도 NHPP에 의한 모델은 모델의 간결성이 높고 적용성이 좋아 다수의 연구자에 의해 연구되었다.

(2) 분산 개발환경에 대한 SDE에 의한 SRGM[10] - SDE 모델

$$E[N_c(t)] = \left[1-\left\{\sum_{i=1}^{n} P_i e^{-b_i t} + \sum_{j=1}^{m} P_{n+j}(1+b_{n+j}t)e^{-b_{n+j}t}\right\}.e^{\frac{\sigma^2}{2}t}\right] \quad (2)$$

$N_c(t)$는 시험시간 t마다 발견.수정된 총 fault 수를 의미하며, 식 (2)는 기대치를 나타낸다. 또한 m_0는 시험시작전 잠재하고 있는 총 기대 fault 수를 $b_l(l=1,2,3,.....n+m)$은 l번째 컴포넌트에 대한 잔존 fault 1개당 fault 발견율을 파라미터 $P_l(l=1,2,3,.....n+m)$은 l번째 컴포넌트에 대한 중복, 다시 말해서 각 소프트웨어 컴포넌트의 규모, 시험역량, 시험 툴 등에 영향을 주는 시험에 관한 중요도를 표시한다. 또한 σ는 백색잡음의 크기로 표시되는 정수 파라미터이다.

과거의 NHPP 모델은 시험단계에 의한 fault 발견 과정을 이산과정(離散過程)으로 취급하였다. 다시 말해서 대상 소프트웨어 시스템이 $1*10^6$ 코드 행수 이상의 대규모인 경우는 시험단계의 발견된 fault 수에 대응한 크기에 대해 각 개인의 디버깅 작업의 수정, 제거를 통한 fault 수의 변화량은 시험 종료까지 개발단계에 작업한 총 fault에 비례하여 충분히 적은 것으로 고려하였다. 이러한 경우에 fault 발견과정을 이 값이 연속적으로 변동되는 과정을 모델화를 거치는 것도 가능한 것으로 고려하였다.

식 (1) 및 (2)에 의해 기술된 2개의 모델은 분산 개발환경을 대상으로 하여 기존의 컴포넌트가 n개, 신규개발 컴포넌트가 m개를 이용하는 경우를 가정하고 있다. 특히 본 모델은 과거 제안된 소프트웨어 신뢰도 성장모델[11-12]에 비해 중복 파라미터 $P_l(l=1,2,3,.....n+m)$의 값을 대응한 지수형 성장곡선부터 S-자형 성장곡선까지 누적발견 fault 수에 관해 광범위하게 특성곡선을 형성할 수 있는 것으로부터 기존의 SRGM에 비해 정확한 신뢰성 평가척도의 추정이 가능하고 프로젝트 관리자에 의해 적용모델의 선택작업이 노력 결과에 대해 확인이 가능하다. 즉, 위와 같은 추세에 따라 최근에 분산 개발환경에 대한 소프트웨어 신뢰성 평가 방법이 활발히 연구되고 있다.[19-20]

본 논문에서는 분산 개발환경에 관한 소프트웨어 신뢰성을 정량적으로 평가하는데 있어서 매우 현실적인 fault 발견과정을 고려한 식 (2)에 의한 SDE 모델에 근거한 최적 배포문제를 논한다. 다시 말해서 소프트웨어 컴포넌트를 통합한 후 최종 단계에 위치하고 있는 통합시험 단계는 시스템 전체 규모의 크기로부터 본 논문은 SDE 모델을 운용 시스템에 의한 fault 발견과정을 근사적으로 표현이 가능한 모델을 고려하였다. 특히, 시험단계 및 소프트웨어의 운용단계 비용이나 총 소프트웨어 비용의 확률변수를 다루어 총 소프트웨어 비용 및 최적 배포시기의 통계적 신뢰구간에 의한 존재 범위를 이용한 총 소프트웨어 비용을 최소로 하는 최적 배포시

기의 추정을 수치를 이용하여 보였다.

Ⅲ. 소프트웨어 최적 배포 문제

고 신뢰성 소프트웨어 제품을 개발하는데 있어서 각 개발공정에 의한 관리기술이 필요하다. 특히, 소프트웨어 개발 공정의 최종 단계인 시험단계에 대해서는 고 신뢰성을 실현하여야 하며 배포된 품질을 파악하는 시험을 행한다. 이런 시험이 종료되는 시점에서는 실제 사용자에 의한 운용단계로의 이행을 수행한다. 즉, 소프트웨어 제품을 시장에 출하하게 된다.

소프트웨어 성능이나 신뢰성은 적용된 시험기술이나 시험 툴의 질에 의해 시험비용이나 실시된 시간에 관계된다. 소프트웨어 시험공정에 의해 고 신뢰성을 실현하기 위해서는 장시간의 시험을 실시하여야 하는바 소프트웨어 신뢰성을 향상시키는 것이 시험에 소요되는 비용이나 작업에 많은 비용이 든다. 이것에 대해 시험비용의 억제를 위해서는 충분한 시험을 수행한 소프트웨어 제품을 조기에 배포하고 이 결과 소프트웨어 내의 잔존 fault가 증가시 소프트웨어 출하 후 유지보수 비용이 증가하게 된다. 여기서 신뢰성과 비용의 trade-off 관계를 고려하여 소프트웨어를 출하할 때 다양한 측면을 고려해야 한다. 이러한 의사 결정 문제를 소프트웨어 최적 배포 문제(optimal software release problem)라고 한다.[13-14]

일반적으로 운용단계에서의 fault 수정에 수반되는 유지보수 비용은 시험공정에 의한 보수비용에 비해 크다. 다시 말해서 소프트웨어 최적 배포문제에 대해 시험공정 및 운용단계에 의한 유지보수 비용을 고려하여 총 소프트웨어 비용을 최소로 하는 최적 배포시각을 구하는 것을 고려한다.[17-18]

본 논문에서는 분산 개발환경에 의해 개발된 소프트웨어 시스템에 대한 비용 평가 기준을 소프트웨어 부품(component)에 착안하여 비용 요인을 고려한 총 소프트웨어 비용을 채용하고 이것을 최소화 하는 최적 배포시기를 결정하는 문제에 대해 논한다.

3.1 소프트웨어 비용

본 절에서는 2장에서 논의한 SDE 모델을 사용하여 분산 개발된 소프트웨어 개발 관리상의 문제를 응용한 한 예로써 시험단계 및 운용단계의 비용 요인을 거론한 총 소프트웨어 비용을 평가기준으로 하여 최적의 출하시기를 결정하는 최적 배포시기 문제를 논한다.

총 소프트웨어 비용을 정식화하는데 있어서 아래의 파라미터를 정의한다.

- *definition of parameters*

c_{1i} : 컴포넌트 i의 단체시험에 의해 발견된 fault 1개당 수정비용($c_{1i}>0$)

c_{2i} : 컴포넌트 i의 단체시험의 단위 시간당 소요된 시험비용($c_{2i}>0$)

c_{1c} : 통합 시험에 의해 발견된 fault 1개당 수정비용($c_{1c}>0$)

c_{2c} : 통합 시험에 단위시간당 소요된 시험 비용($c_{2c}>0$)

c_{3c} : 운용 후 발견된 fault 1개당 소요되는 유지보수 비용($c_{3c}>0, c_{3c}>c_{1i}, c_{3c}>c_{1c}$)

여기서 각 서브시스템에 의한 누적 발견 fault 수 데이터의 성장곡선 형상에 의해 아래에 표시한 NHPP 모델에 의한 지수형 SRGM[15, 21]과 지연 S-자형 SRGM[16]을 이용하여 신뢰성 평가법의 적용을 제안하였다.[7]

• 지수형 SRGM

$$H_e(t)=a(1-e^{-bt})(a,b>0) \tag{3}$$

• 지연 S-자형 SRGM

$$H_d(t)=a\{1-(1+bt)e^{-bt}\}(a,b>0) \tag{4}$$

여기서 $H_e(t)$ 및 $H_d(t)$는 NHPP 모델에 의한 평균치 함수로 시간 구간$(0,t)$에 의해 발견된 기대 누적 fault 수를 표시한다. 또한 a 및 b는 시험 개시 전 잠재하고 있는 총 기대 fault 수 및 fault 발견율을 표시하는 정수 파라미터이다.

한편 각 컴포넌트의 통합 시험단계에 지연의 영향에 의해 예정된 시기에 운용이 불가능한 일이 발생된 사용자에 대해 penalty(보상) 등을 고려할 필요성이 있으며, 단체시험(unit test) 종료시점이 통합시험(integration test) 개시시각을 초과하는 경우는 penalty 비용을 부과하는 것도 고려하여야 한다. 여기서 개발된 소프트웨어의 종류나 규모에 의존하는 것도 고려하여야 하는바 일반적으로 각 소프트웨어 컴포넌트의 통합시험 단계에 대한 지연이 진행됨에

따라 이에 대한 penalty는 지수 함수적으로 증가되는 것으로 가정하여야 한다. 또한 수학적 취급상의 이용성을 고려한 결과, 컴포넌트(component) $i(i=1,2,3,.....,n+m)$의 단체시험으로부터 통합시험 공정에 투입되는 지연시간$(t_i - t_{di})$에 대한 penalty 비용 함수를 아래와 같이 정식화 한다.

$$G_i(t_i)=\begin{cases} c_{3i}\left\{e^{k_i(t_i-t_{di})}-1\right\} (t_i > t_{di}) \\ 0 \quad (t_i \le t_{di}) \end{cases} \tag{5}$$

t_{di}는 컴포넌트 i의 단체시험 개시시각으로부터 통합시험 개시시각까지의 시간$(t_{di}>0)$을 표시하고 t_i는 컴포넌트 i의 단체시험 종료시각을 표시한다. $c_{3i}(>0)$ 및 $k_i(>0)$는 여러 컴포넌트 i의 단위시간당 소요되는 지연 비용 및 지연에 의한 통합 시험 공정에 대한 영향도의 합을 표시한다. 여기서 아래의 각 컴포넌트의 시험에 소요되는 기대 비용의 식을 얻을 수 있다.

$$C_i(t_i)=c_{1i}H_i(t_i)+c_{2i}t_i+G_i(t_i) \tag{6}$$
$$(i=1,2,3,...n,..n+m)$$

$H_i(t_i)$는 i번째 컴포넌트에 대해 적용된 NHPP에 의한 지수형 SRGM부터 지연 S-자형 SRGM의 평균치 함수를 표현한다. 이것에 의해 총 소프트웨어 비용은

$$Cost(N_c(t_c),t_c)= \tag{7}$$
$$\sum_{i=1}^{n+m} C_i(t_i)+c_{1c}N_c(t_c)+c_{2c}t_c+c_{3c}\{m_0-N_c(t_c)\}$$

로 정식화 된다. 여기서 t_c는 통합시험 단계에 의한 시험시각을 표시한다. $N_c(t_c)$는 확률변수의 파라미터로 $Cost(N_c(t_c),t_c)$는 확률변수로 사용된다. 식 (7)을 변형하면

$$Cost(N_c(t_c),t_c)=\sum_{i=1}^{n+m} C_i(t_i)+c_{2c}t_c- $$
$$(c_{3c}-c_{1c})N_c(t_c)+m_0c_{3c} \tag{8}$$

가 된다. 여기서 $N_c(t_c)$의 분포함수가 식 (2)의 SDE 모델의 경우는

$$P_r[N_c(t_c)\le n]= \Phi\left(\left\{\log\frac{m_0}{m_0-n}+\log\left[\sum_{i=1}^{n}P_ie^{-b_it_c}+\sum_{j=1}^{m}P_{n+j}(1+b_{n+j}t_c).e^{-b_{n+j}t_c}\right]\right\}/ \sigma\sqrt{t_c}\right) \tag{9}$$

로 되고 이 포도함수를 이용하여 $Cost(N_c(t_c),t_c)$의 분포함수를 고려한다. $\Phi(.)$는 표준 정규분포함수를 표현하며,

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp\left(-\frac{\zeta^2}{2}\right)d\zeta \tag{10}$$

로 정의된다. 식 (8)에 의해

$$N_c(t_c)=\left\{\sum_{i=1}^{n+m}C_i(t_i)+c_{2c}t_c+m_0c_{3c}-Cost(N_c(t_c),t_c)\right\} \tag{11}$$
$$/(c_{3c}-c_{1c})$$

이 된다. 식 (9)에 식 (11)을 대입하여 변형한다. 여기서

$$C=-n(c_{3c}-c_{1c})+(c_{2c}t_c+m_0c_{3c}) \tag{12}$$

와 $Cost(N_c(t_c),t_c)$의 분포함수는

$$P_r[Cost(N_c(t_c),t_c)\le C]= 1-\Phi\left(\left\{\log\left[m_0(c_{3c}-c_{1c})/\left\{C-\left(\sum_{i=1}^{n+m}C_i(t_i)+c_{2c}t_c+m_0c_{1c}\right)\right\}\right]+ \log\left[\sum_{i=1}^{n}P_ie^{-b_it_c}+\sum_{j=1}^{m}P_{n+j}(1+b_{n+j}t_c)e^{-b_{n+j}t_c}\right]\right\}/\sigma\sqrt{t_c}\right) \tag{13}$$

가 된다. 한편, 총 기대 소프트웨어 비용은 식 (7)의 기대치를 통해 식 (14)로 표현된다.

$$E[Cost(N_c(t_c),t_c)]=\sum_{i=1}^{m+n}C_i(t_i)+c_{1c}E[N_c(t_c)]+ c_{2c}t_c+c_{3c}\{m_0-E[N_c(t_c)]\} \tag{14}$$

3.2 소프트웨어 비용의 신뢰구간과 최적배포 시기

본 논문에서는 소프트웨어 비용을 확률 변수로 취급하여 비용의 분포함수(cost distribution function)의 값이 $(1-0.01\alpha)/2$가 되는 비용으로부터 $(1+0.01\alpha)/2$까지 비용을 구하고 소프트웨어 비용의 $\alpha\%$ 구간을 구하는 것이 가능하다. 이 구간을 소프트웨어 비용의 $\alpha\%$ 신뢰구간(confidence interval)이라 한다. 여기서 소프트웨어의 $\alpha\%$ 신뢰구간의 상.하한을 $C_U(t_c)$ 및 $C_L(t_c)$로 정하고 이를 각각 표시하면

$$C_U(t_c)=\sum_{i=1}^{n+m}C_i(t_i)+c_{2c}t_c+m_0c_{1c}+ \left\{m_0(c_{3c}-c_{1c})/\exp\left[\sigma\sqrt{t}\,\Phi^{-1}\left(\frac{1+0.01\alpha}{2}-\right.\right.\right. \left.\left.\left.\log\left[\sum_{i=1}^{n}P_ie^{-b_it_c}+\sum_{j=1}^{m}P_{n+j}(1+b_{n+j}t_c).e^{-b_{n+j}t_c}\right]\right)\right]\right\} \tag{15}$$

$$C_L(t_c)=\sum_{i=1}^{n+m}C_i(t_i)+c_{2c}t_c+m_0c_{1c}+\left\{m_0(c_{3c}-c_{1c})/\exp\left[\sigma\sqrt{t}\,\Phi^{-1}\left(\frac{1-0.01\alpha}{2}\right)-\log\left[\sum_{i=1}^{n}P_ie^{-b_it_c}+\sum_{j=1}^{m}P_{n+j}(1+b_{n+j}t_c).e^{-b_{n+j}t_c}\right]\right]\right\} \quad (16)$$

이 된다. 그 다음 최적 배포시기를 논하면 최적 배포시각을 T^*라고 하면 T^*는 식 (14)의 총 기대 소프트웨어 비용을 최소로 하는 시각 $t_c=T^*$를 구해 얻을 수 있다.

한편 본 논문에서는 $\alpha\%$ 신뢰구간의 $C_U(t)$와 $C_L(t)$를 최소로 하는 시각 즉, ($t_c=T_U^*$ 및 $t_c=T_L^*$)를 구하고 $\alpha\%$ 신뢰구간에 대해서 최적 배포시각의 범위를 구하면 된다. 여기서 T_U^*는 $C_U(t_c)$를 최소로 하는 시각, T_L^*은 $C_L(t_c)$를 최소로 하는 시각이다.

Ⅳ. 적용 및 결과 분석

실제 시험단계(통합시험)에 의해 관측된 데이터(그림 1 참조)를 적용한 수치의 예를 보였다. 여기에 표시된 수치 예는 실제 개발된 프로젝트 데이터에 의한 것이며, 월별 시스템의 주요 소프트웨어 컴포넌트에 관한 고장 발생 현황은 표 1과 같다.

본 논문에 사용된 데이터는 분산 개발 환경에 의해서 개발된 소프트웨어로써 총 개발규모가 1,680만 라인 규모이며, 컴포넌트는 신규개발(n=68) 및 기 개발(m=99)된 부분을 변경, 유용(약 67%)한 167개의 소프트웨어 컴포넌트로 구성된 대형 소프트웨어 시스템의 시험단계로부터 수집된 것이다. 여기서 시험기간 t_c는 측정 단위가 주(week)이다.

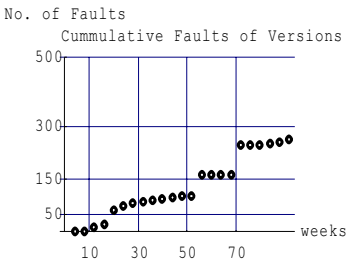

그림 1. 시스템 전체 누적 fault 수(실측치)

표 1. 소프트웨어 주요 컴포넌트별 fault 발생(월별)

(단위 : 건)

components	detection_faults				
	3월	4월	5월	6월	7월
dynamic_routing(15)	15	-	-	-	-
fwd_plane(32)	13	4	13	3	-
sw_test(21)	1	4	10	6	-
embedded_sw(30)	15	-	3	6	4
μ_code(13)	3	4	-	5	1

또한 본 논문에서 다루는 소프트웨어 컴포넌트의 신뢰성 평가 기준은 크게 3개 부분으로써 QSS(QoS Switching System)의 문제점 관리 시스템(bug tracking system으로 사용중)인 ECR(Engineering Change Request)로부터 이용 가능한 고장 데이터를 이용, 시스템 전체에 영향을 주는 요인을 고려하였다. 즉, 아래와 같이 크게 3개 부분으로 구분하였다.

(1) 각 컴포넌트에 대한 fault 중요도(severity)
(2) fault 수정(해결)자(assign to ~)
(3) fault Reporter

4.1 신뢰성 평가 예

상기 데이터를 적용한 모델 파라미터의 추정 결과는 아래와 같다.

$$\widehat{m_0}=664.236, \hat{b_i}=0.0261, \hat{b_j}=0.015, \hat{\sigma}=0.045, b_{n+j}=0.0205$$

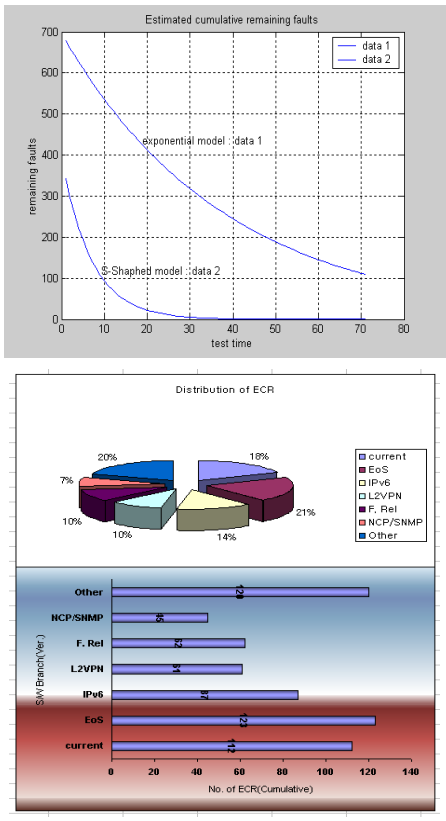

그림 2. 추정된 잔존 fault 수의 서브 경로(상) 및 S/W 버전별 고장 분포(하) 현황

또한 중첩 파라미터 $P_l(l=1,2,3,....,n+m)$에 의한 최우 추정치의 최대는

$\sum_{i=1}^{n} P_i = 0.2, \sum_{j=1}^{m} P_{n+j} = 0.8$의 값을 적용하였다.

여기서 추정된 잔존 fault 수의 서브 경로 $\widehat{m_c(t_c)}(=\widehat{m_o}-\widehat{N_c(t_c)})$를 그림 2에 표시하였다. 또 그림 2의 아래(하) 그림은 각 소프트웨어 주요 버전별 고장 분포와 발생횟수를 표시한다. 이 결과에 의하면 고장분포는 자체 개발중인 EoS(Ethernet over Sonet) 기능과 IPv6, 초기 QoS Switching System(QSS)의 핵심 기본기능인 Routing/Packet Forwarding 기능(Current) 등에 많은 고장과 변경이 가해지고 있음을 데이터 분석결과 밝혀졌다.

그밖에 현장 적용(F. Rel)을 위한 버전도 전체 고장의 약 10% 정도를 차지하고 있다.

또한, 신뢰성 평가의 한 예로 소프트웨어 고장의 발생빈도의 특성을 조합한 유용한 평균 고장 발생 시간 간격(MTBF: mean time between software failures)의 척도에 의한 순간 MTBF(instantaneous MTBF, $MTBF_I$)와 시험개시 시점을 고려한 누적 MTBF(cumulative MTBF, $MTBF_C$)는 식 (17), (18)에 의해 근사적으로 표현된다.[12]

$$MTBF_I(t_c) = \frac{1}{E[\frac{dN_c(t_c)}{dt_c}]} \quad (17)$$

$$MTBF_c(t_c) = \frac{t_c}{E[N_c(t_c)]} \quad (18)$$

또 소프트웨어 fault의 누적 및 순간 발생률은 그림 3에, 식 (17) 및 (18)에 의해서 추정된 MTBF와 누적 fault 발생 추정치는 그림 4에 각각 표시하였다.

그림 3으로부터 시험 진도에 의한 순간 MTBF 및 누적 MTBF의 값이 증가되는 것을 확인할 수 있다. 다시 말해서 소프트웨어 신뢰도성장이 향상되는 것을 이것을 통해 알 수 있다. 그밖에 시험단계에서 발생되는 fault 데이터를 적용하여 2개 모델에 적용, 추정한 축적 고장 데이터는 그림 4와 같다. (그림 4의 아래(하) 그림에 대한 data 1, 2는 exponential 및 S-shaped model의 추정치임)

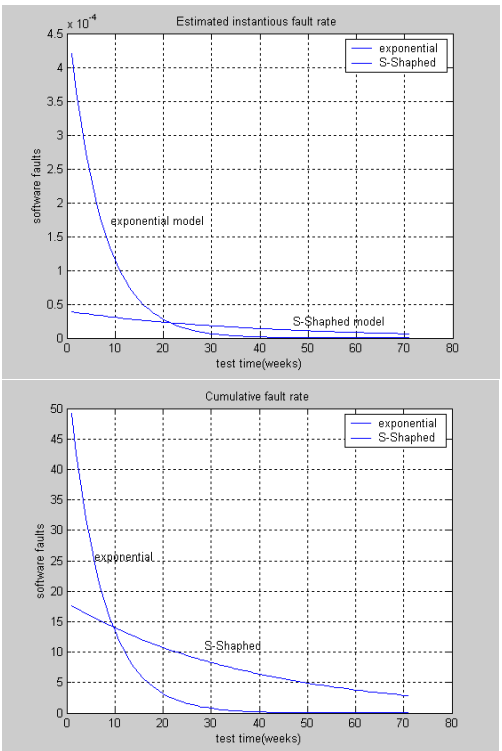

그림 3. 추정된 소프트웨어 fault 발생률 (상: 순간발생률, 하: 누적발생률)

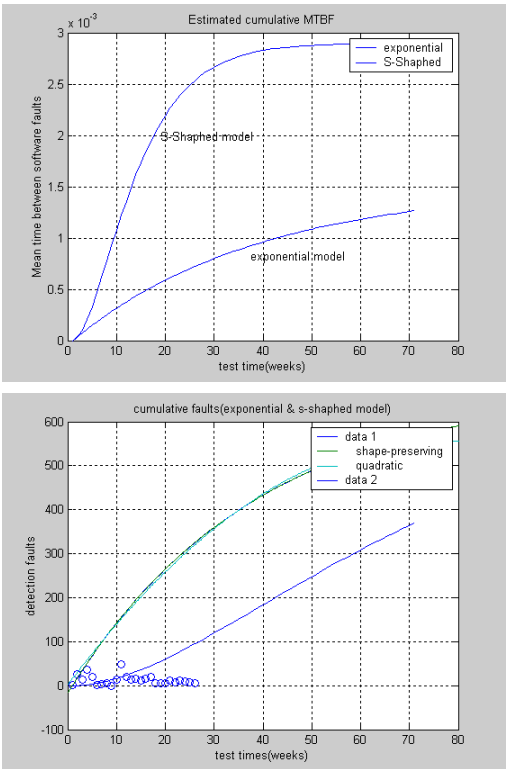

그림 4. MTBF 및 소프트웨어 결함 검출 추정치, 실측치

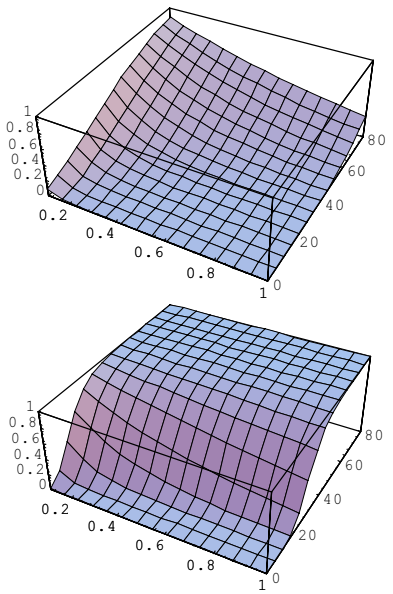

그림 5. 신뢰도[$\widehat{R_c(t)}$]추정치(상 : 지수형, 하 : S-자형[2차 Gamma 모델])

또 NHPP(Non-Homogeneous Poisson Process) 모델로부터 소프트웨어 신뢰성 평가에 의한 정량적 척도를 도출하기 위한 방법으로써 시험시각 t에 의한 소프트웨어 내(內)의 기대잔존(期待殘存) fault 수의 기대치(그림 1 참조)와 시각 t 마다 시스템 운용을 행할 때 시간 구간 $(t,t+x]t>0,x>0)$에 의한 소프트웨어 고장이 발생하지 않을 조건부 확률로 표현할 수 있는데, 이것은 아래 식 (12)와 같이 표현하며, 이것을 소프트웨어 신뢰도라고 부른다.[19]

$$R_c(x|t)=\exp[H_c(t)-H_c(t+x)] \qquad (12)$$

위의 함수식에 의거하여 신뢰도 평가추이(trend analysis)를 예측해 보면 그림 5와 같다.

3차원 그래프로 나타낸 그림 5의 y축은 신뢰도(0.1 ~ 1.0)를 표현하고 x축 및 z축은 운용시간(0.1 ~ 1)과 시험시각(0 ~ 80주)을 의미한다.

4.2 최적배포 문제

3.1항의 총 기대 소프트웨어 비용을 이용하여 소프트웨어 최적 배포 문제의 수치 예를 표시한다. 여기서는 하나의 예로써 아래의 파라미터들을 설정하여 최적 배포시기를 구하면

$$c_{11}=1,c_{12}=1,c_{13}=1,c_{14}=1,c_{15}=1,c_{16}=1,c_{17}=2,c_{18}=1,c_{19}=2$$
$$c_{21}=2,c_{22}=2,c_{23}=2,c_{24}=2,c_{25}=2,c_{26}=2,c_{27}=4,c_{28}=2,c_{29}=4$$
$$c_{1c}=10,c_{2c}=20,c_{3c}=50$$

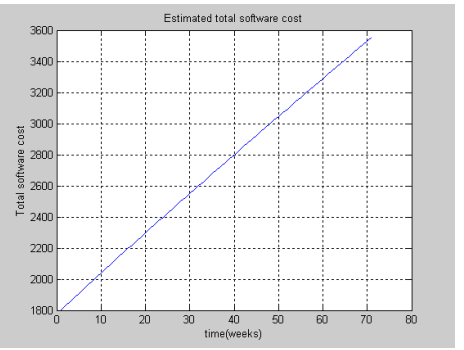

그림 6. 소프트웨어 비용의 샘플 경로

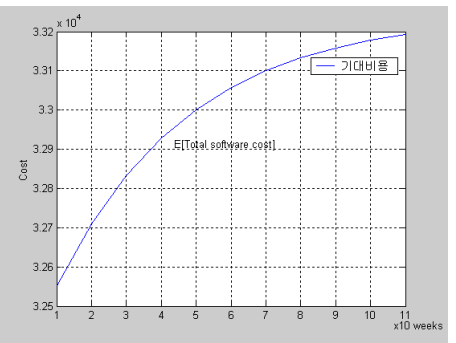

그림 7. 추정된 총 기대 소프트웨어 비용

여기서 4.1항의 신뢰성 해석 결과에 의해 추정된 소프트웨어 비용의 샘플 경로를 그림 6에 표시하였다. 그 다음에 추정된 총 기대 소프트웨어 비용의 시간 변화의 모양을 그림 7에 나타내었다. 그림 7로부터 최적 배포 시기는 $T^*=132.847$로 추정되고 이때 총 기대 소프트웨어 비용은 $3.32*10^4$ 이었다. 또한 소프트웨어 비용을 확률 변수로 취급하여 소프트웨어 비용의 분포함수의 값이 0.05가 되는 비용과 0.95가 되는 비용을 구하면 90% 신뢰구간을 구하는 것이 가능하다.

그림 6에 의해 소프트웨어 비용의 90% 신뢰구간의 변화가 표시되는 $C_U(\widehat{t_c})$ 및 $C_L(\widehat{t_c})$에 의한 소프트웨어 비용이 최소가 되는 시각은 각각 ($T_U^*=165$ 및 $T_L^*=160$)으로 확인되었다. 다시 말해서 90% 신뢰 구간에 의한 소프트웨어 비용의 존재 범위는 각각

$$C_U(T_U^*)=4.78*10^4,\ C_L(T_L^*)=4.75*10^4$$

로 판정되었다. 프로젝트 개발 관리자는 이러한 정보에 의해 시험단계를 종료하고 사용자에게 소프트웨어를 인도하는 최적 시기를 정량적으로 파악이 가능하다. 특히, 통계적 신뢰구간에 의한 존재 범위

를 사용하여 현실적인 최적 배포시기 및 소프트웨어 비용의 산출이 가능하다.

V. 결 론

본 논문에서는 확률미분방정식을 도입한 분산 개발환경에 대한 SRGM으로부터 시험 단계에 발견된 소프트웨어 fault 수에 관한 데이터 해석을 행한 신뢰성 평가를 실시한 후에 운용단계로 이행하기 위한 최적 시기를 결정하기 위한 분산 개발환경을 대상으로 한 소프트웨어의 최적 배포 문제를 논하였다. 다시 말해서 총 기대 소프트웨어 비용을 최소로 하는 시각을 구하고 최적 배포시각을 결정하기 위한 평가 기준에 대해 소프트웨어 비용을 확률변수로 취급하여 소프트웨어 비용의 $\alpha\%$ 신뢰구간을 사용하고 최적 배포시각 및 총 기대 소프트웨어 비용의 존재 범위를 확인, 소프트웨어 개발 관리상의 중요한 현실적인 최적 배포시기를 구하는 것이 가능하다.

과거에는 소프트웨어 비용의 기대치를 최소로 하는 확정적인 최적 배포시기를 구하고 있으나 본 논문에서는 소프트웨어 fault 수의 분포가 대수 정규분포에 의해서 소프트웨어 비용의 분포가 구분됨으로 인해 소프트웨어 비용을 최소로 하는 $\alpha\%$의 확실한 최적 배포시기를 구하는 것이 가능하다.

한편 각 소프트웨어 컴포넌트의 통합 시험 단계에 대한 납기 지연의 penalty 비용을 고려하여 최적 배포 문제의 해를 얻을 수 있는 현실적인 최적 출하시기를 산출해 내는 것이 가능하며, 이러한 규정의 납기와 비교를 통해 개발 관리자의 의사 결정에 일조하고 있다.

본 논문에서는 소프트웨어 비용의 함수에 대해서 기본적인 요인을 다루고 논의하였으며, 소프트웨어 신뢰성 평가 척도와는 별도로 요인을 취급하는 평가를 향후 과제로 제시한다.

참 고 문 헌

〔1〕 松本政雄, 小山田正史, 松尾谷徹, ソフトウェア開發檢証技法, 電子情報通信學會, 東京, 1997.

〔2〕 R. S. Pressman, Software engineering: A Practitioner's Approach(4th ed.), McGraw-Hill, New York, 1982.

〔3〕 A. Umar, Distributed Computing and Client-Server Systems, Prentice Hill, New Jersey, 1993.

〔4〕 L. T. Vaughn, Client/Server System Design and Implementation, McGraw-Hill, New York, 1994.

〔5〕 高橋宗雄, クライント/サーバシステム開發の工數見積り技法-工數見積りモデルの適用法, ソフト・リサーチ・センター, 東京, 1998.

〔6〕 赤羽豊和, クライントサーバ・システムのテスト技法, ソフト・リサーチ・センター, 東京, 1998.

〔7〕 山田 茂, ソフトウェア信賴性モデル-基礎と應用, 日科技連出版社, 東京, 1994.

〔8〕 L. Arnold, Stochastic Differential Equations - Theory and Applications, John Willy & Sons, New York, 1974.

〔9〕 山田 茂, 田村慶信, 木村光宏, "分散開發環境を考慮したソフトウェア信賴度成長モデルに關する考察", 新學論(A), vol. J.82-A, no. 9, pp. 1446-1453, sept, 1999.

〔10〕 田村 慶信, 木村 光宏, 山田 茂, "分散開發環境に對するソフトウェア信賴度成長モデル : 確率微分方程式アプローチとその推定", 日本應用水理學會論文誌, vol. 11, no. 3, pp. 121-132, Sept. 2001.

〔11〕 M. Lyu(ed.), Handbook of Software Reliability Engineering, McGraw-Hill, New York, 1996.

〔12〕 S. Yamada, M. Kimura, H. Tanaka, and S. Osaki, "Software reliability measurement and assessment with stochastic differential equations", IEICE Trans. Fundamentals, vol. E77-A, no. 1, pp. 109-116, Jan. 1994.

〔13〕 S. Yamada, S. Osaki, "Cost-reliability optimal release polices for a software system", IEEE Trans. Reliab., vol. R-34, no. 5, pp. 422-424, Dec. 1985.

〔14〕 S. Yamada, S. Osaki, "Optimal software release polices with simultaneous cost and reliability requirements", Eur. J. Oper. Res., vol. 31, no. 1, pp. 46-51, July, 1987.

〔15〕 A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures", IEEE Trans. Reliab., vol. R-28, no. 3, pp. 206-211, Aug. 1979.

〔16〕 S. Yamada, M. Ohba, and S. Osaki, "S-Shaped reliability growth modeling for software error

detection", IEEE Trans. Reliab., vol. R-32, no. 5, pp. 475-478, 484, Dec. 1983.

〔17〕 H. S. Kim, S. Yamada, D. H. Park, "Bayesian Approach to Optimal Release Policy of Software System", IEICE Trans. on Fundmentals, Vol. E88-A, No. 12, pp. 3618-3626, Dec. 2005.

〔18〕 C. Y. Huang and Michael R. Lyu, "Optimal Release Time for Software Systems Considering Cost, Testing-Effort, and Test Efficiency", IEEE Trans. on Reliability, Vol. 54, No. 4, pp. 583-591, Dec. 2005.

〔19〕 이재기외 5, "공동 개발환경 소프트웨어의 신뢰도 평가 방법", 7th CEIC2005 Proceeding, pp. 181-187, Dec. 2005.

〔20〕 田村 慶信, 山田 茂, 木村 光宏, "オープンソース共同開發環境に對するソフトウェア信賴性評價に關する考察", 電子情報通信學會論文誌 A, Vol. J88-A, No. 7, pp. 840-847, July 2005.

〔21〕 J.K. Lee, S.S. Nam, C.B. Kim, "An Evolution of Software Reliability in a Large Scale Switching System: using the software change management data", KICS, Vol.29. No. 4A, pp. 399-414, March .2004.

〔22〕 J.K. Lee, G..O. Lee, C.B. Kim, "Software Reliability Growth Models considering an Imperfect Debugging environments", KICS, Vol. 29, No. 6A, pp. 589-599, June. 2004.

이 재 기 (Jae-ki Lee) 정회원

1985년 2월 서울산업대학교 전자공학과 졸업

1989년 5월 청주대학교 전자공학과 석사

2004년 2월 공주대학교 전기전자정보공학과 박사

1983년 3월~현재 한국전자통신연구원 책임연구원

<관심분야> 소프트웨어 신뢰도, 검증, 테스팅, BcN

남 상 식 (Sang-sik Nam) 종신회원

1981년 2월 단국대학교 전자공학과 졸업

1983년 2월 단국대학교 전자공학과 석사

1996년 3월 단국대학교 전자공학과 박사

1999년 3월~현재 한국전자통신연구원 책임연구원

<관심분야> ATM, Signal Integrity, BcN