

수명분포가 지수화-지수분포를 따르는 소프트웨어 신뢰모형 특성에 관한 연구

김희철* · 문송철**

A Study on the Characteristics of Software Reliability Model Using Exponential-Exponential Life Distribution

Hee Cheul Kim* · Song Chul Moon**

Abstract

In this paper, we applied the shape parameters of the exponentialized exponential life distribution widely used in the field of software reliability, and compared the reliability properties of the software using the non-homogeneous Poisson process in finite failure. In addition, the average value function is also a non-decreasing form. In the case of the larger the shape parameter, the smaller the estimated error in predicting the predicted value in comparison with the true value, so it can be regarded as an efficient model in terms of relative accuracy. Also, in the larger the shape parameter, the larger the estimated value of the coefficient of determination, which can be regarded as an efficient model in terms of suitability. So, the larger the shape parameter model can be regarded as an efficient model in terms of goodness-of-fit. In the form of the reliability function, it gradually appears as a non-increasing pattern and the higher the shape parameter, the lower it is as the mission time elapses. Through this study, software operators can use the pattern of mean square error, mean value, and hazard function as a basic guideline for exploring software failures.

Keywords : Mean Squared Error, Exponential-Exponential Distribution, Reliability, Determination Coefficient, Non-Homogeneous Poisson Process.

Received : 2020. 06. 11. Revised : 2020. 06. 25. Final Acceptance : 2020. 06. 29.

* Professor, Namseoul Univ., Dept. of Industrial & Management Engineering, e-mail : kim1458@nsu.ac.kr

** Corresponding Author, Professor, Namseoul Univ. Dept. Computer Software, 91 Daehak-ro, Seonghwan-eup, Seobuk-gu, Cheonan-si, Chungcheongnam-do, 31020, Korea, Tel : +82-2-820-5454, e-mail : moon@nsu.ac.kr,

1. 서론

소프트웨어 시스템은 산업경영의 중요한 지적요소 중에서 기본요소로 간주될 수 있다. 이러한 시스템은 정확성(Correctness)과 신뢰성(Reliability)을 확보할 수 있다면 소프트웨어 사용자에게 고품질의 서비스를 제공할 수 있다. 따라서 소프트웨어 유지 및 보수는 소프트웨어 시스템의 정확성과 신뢰성을 유지하는데 있다. 이러한 유지 및 보수 관련 엔지니어링 연구인 소프트웨어 신뢰 성장모형(Software reliability growth model)이 제안되어 왔다. 이러한 연구는 남아 있는 고장의 수, 고장강도 등의 신뢰성 속성 요인을 이용하여 고장강도 속성 및 평균값 함수를 예측하는 비동질 포아송 과정(Non-homogeneous Poisson process: NHPP)을 따르는 소프트웨어 신뢰성 모형이 개발되었다(Song and Pham, 2017).

소프트웨어 신뢰성 모형은 소프트웨어의 신뢰성, 남은 고장 수, 고장강도, 소프트웨어 개발 총비용 등을 예측 및 검정하는 데 사용된다. 이러한 환경에서 Yamada와 Osaki는 최우추정법을 이용하여 평균값 함수의 결과를 예측할 수 있음을 강조하였고 평균값 함수의 신뢰구간을 나타내는 그래프를 제시하여 신뢰성 특성을 설명하였다(Yamada and Osaki, 1985; Teng and Pham, 2006). 이 분야에서 기본모형인 지수분포를 이용하여 개발된 결합 탐색비용은 상수(위험함수)인 강도함수로 구성되었다(Goel and Okumoto, 1979). 그리고 Kim은 수명분포가 Burr-XII와 Type-2 Gumbel 분포를 이용한 신뢰특성을 연구하기도 했다(Kim, 2019). 본 연구에서는 소프트웨어 신뢰성 분야에서 널리 사용되는 지수화-지수분포를 따르는 수명분포의 형상모수를 적용하고 유한고장수를 가진 비동질적인 포아송 과정을 활용한 소프트웨어의 신뢰성 모형 특성을 제시하였다.

2. 유한고장 NHPP 소프트웨어 신뢰성 모형

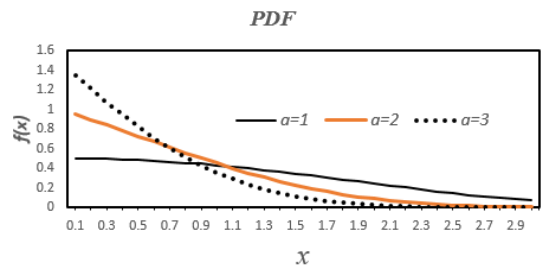
2.1 지수화-지수분포

소프트웨어 신뢰성 분야에서 널리 사용되는 모형 중에서 와이블 지수화 분포 중에서 특수한 형태가 지수화-지수분포이고 이 분포의 확률밀도함수와 누적분포함수는 다음과 같다(Oguntunde et al., 2015).

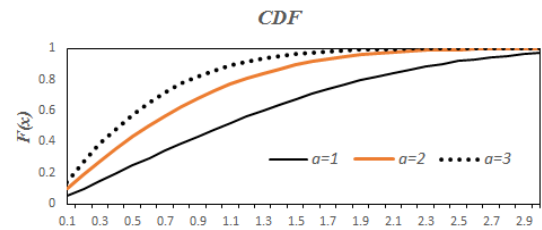
$$f(x) = ab \exp(bx - a e^{bx} + a) \quad (1)$$

$$F(x) = 1 - \exp(-a e^{bx} + a) \quad (2)$$

위 식에서 a 는 형상모수이고 b 는 척도모수이다.



〈Figure 1〉 The Probability Density Function Pattern Case of $b = 0.5$



〈Figure 2〉 Cumulative Density Function Pattern Case of $b = 0.5$

척도모수 $b = 0.5$ 일 경우 확률밀도함수와 누적분포함수의 패턴은 각각 〈Figure 1〉, 〈Figure 2〉와 같다.

2.2 수명분포가 지수화-지수분포를 따르는 NHPP 소프트웨어 신뢰모형

수명분포가 지수화-지수분포를 따르는 NHPP 소프트웨어 신뢰모형 대한 강도함수와 평균값함수는 식 (1)과 식 (2)을 사용하면 다음과 같다(Teng and Pham, 2006).

$$\lambda(t|\theta, b) = \theta f(t) = \theta [ab \exp(bx - a e^{bt} + a)] \quad (3)$$

$$m(t|\theta, \beta) = \theta F(t) = \theta [1 - \exp(-a e^{bt} + a)] \quad (4)$$

유한고장 NHPP 모형에서 테스트 시간 $(0, t]$ 동안 발견되는 결합의 기대 값이 θ 이고 $f(t)$ 는 확률밀도 함수(PDF)이고 $F(t)$ 는 누적분포함수(CDF)를 의미한다. 그리고 식 (3)과 식 (4)를 이용하고 $\theta = \{\theta, a, b\}$ 을

모수공간이라고 표시하면 유한고장 NHPP모형의 로그 우도함수는 다음과 같다[Kim, 2019; Gokhale and Trivedi, 1999].

$$\begin{aligned} \ln L_{NHPP}(\theta | \underline{x}) &= -m(x_n) + \left(\sum_{i=1}^n \ln \lambda(x_i) \right) \quad (5) \\ &= -\theta \left[1 - \exp(-a e^{bx_n} + a) \right] \\ &\quad + \sum_{i=1}^n \ln \left[\theta (a b \exp(bx - a e^{bx_i} + a)) \right] \end{aligned}$$

단, $\underline{x} = (x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n)$.

따라서 식 (5)에서 고장절단모형에서의 형상모수 a 을 고정하고 모수 θ 와 b 에 관한 편미분식은 다음과 같이 유도할 수 있다[Kim, 2017; Gokhale and Trivedi, 1999].

$$\frac{\partial \ln L(\theta | \underline{x})}{\partial \theta} = \frac{n}{\theta} - \left[1 - \exp(-a e^{bx_n} + a) \right] = 0 \quad (6)$$

$$\begin{aligned} \frac{\partial \ln L(\theta | \underline{x})}{\partial b} &= \frac{n}{b} + \sum_{i=1}^n x_i - a \sum_{i=1}^n x_i e^{bx_i} \quad (7) \\ &\quad - \theta a x_n \exp(bx_n - a x_n e^{bx_n} + a) = 0 \end{aligned}$$

따라서 식 (6)과 식 (7)을 사용한 최우추정량 $\hat{\theta}_{MLE}$ 와 \hat{b}_{MLE} 은 다음과 같은 조건식을 만족해야 한다.

$$\frac{n}{\hat{\theta}_{MLE}} = 1 - \exp(-a e^{b_{MLE} x_n} + a) \quad (8)$$

$$\begin{aligned} \frac{n}{\hat{b}_{MLE}} &= - \sum_{i=1}^n x_i + a \sum_{i=1}^n x_i e^{b_{MLE} x_i} \quad (9) \\ &\quad + \theta a x_n \exp(b_{MLE} x_n - a x_n e^{b_{MLE} x_n} + a) = 0 \end{aligned}$$

3. 수명분포가 지수화-지수분포를 따르는 소프트웨어 고장시간 신뢰도 분석

이 장에서 소프트웨어 고장시간 데이터(Failure time data)[Satya et al., 2011]를 이용하여 신뢰모형들의 신뢰성 특징을 비교 분석하고자 한다.

소프트웨어 고장시간 자료는 <Table 1>에 요약되었고 극단 값이 존재하는지를 탐색하기 위하여 본 연구에서는 추세검정을 위하여 박스-플롯(Box-plot)을 이용하였다[Kim, 2017]. 즉, <Figure 3>의 결과에서 중앙값 144.015, 제1사분위수는 80.9, 제3사분위

<Table 1> Failure Time Data

Failure number	Failure time (hours)	Failure time $\times 10^{-2}$ (hours)
1	30.02	0.3002
2	31.46	0.3146
3	53.93	0.5393
4	55.29	0.5529
5	58.72	0.5872
6	71.92	0.7192
7	77.07	0.7707
8	80.90	0.8090
9	101.90	1.0190
10	114.87	1.1487
11	115.34	1.1534
12	121.57	1.2157
13	124.97	1.2497
14	134.07	1.3407
15	136.25	1.3625
16	151.78	1.5178
17	177.50	1.7750
18	180.29	1.8029
19	182.21	1.8221
20	186.34	1.8634
21	256.81	2.5681
22	273.88	2.7388
23	277.87	2.7787
24	453.93	4.5393
25	535.00	5.3500
26	537.27	5.3727
27	552.90	5.5290
28	673.68	6.7368
29	704.49	7.0449
30	738.68	7.3868

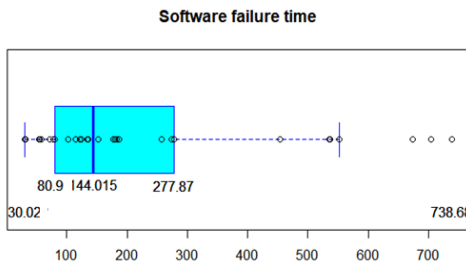
수는 277.87등으로 계산이 되므로 상자그림의 상한과 하한은 각각 다음과 같이 계산된다.

$$277.87 + 1.5 \times (277.87 - 80.9) = 573.325,$$

$$80.9 - 1.5 \times (277.87 - 80.9) = -214.555.$$

따라서 본 자료는 수명자료이기 때문에 하한은 의미가 없지만 상한은 573.325 이므로 상한을 벗어나는 세계의 자료(28, 29, 30번째)가 이상 값(극단 값)으로 간주하여 모수 추정에서 제외되었다[Kim, 2017]. 또한, 모수추정을 용이하게 할 수 있도록 원래의 데이터를 수치변환(Failure time $\times 10^{-2}$)하여 모수추정을 위해

최우추정 법을 적용하고 비선형 방정식의 계산방법은 수치 해석적 방법인 이분법(Bisection method)을 이용하였다. 이러한 계산은 초기 값을 0.01과 3.00을, 허용 한계(Tolerance for width of interval)는 10^{-5} 을 선택하고 수렴 성을 확인되는 충분한 반복 횟수인 100번을 C-언어를 사용하여 모수 추정을 수행하였다. 그 결과는 <Table 2>에 요약되었다.



<Figure 3> Result of Box-Plot

<Table 2> Parameter Estimation, MSE and R^2

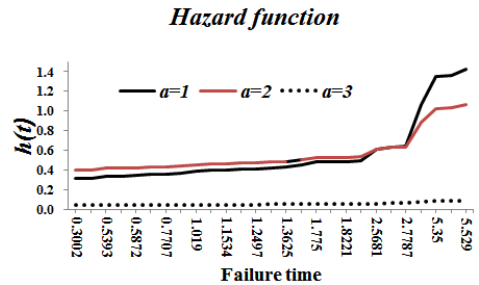
Model	MLE	Model Comparison	
		MSE	R^2
$a = 1$	$\hat{\theta} = 38.5943$ $\hat{b} = 0.2885$	28.0709	0.5887
$a = 2$	$\hat{\theta} = 30.6612$ $\hat{b} = 0.1879$	4.7840	0.9299
$a = 3$	$\hat{\theta} = 28.2757$ $\hat{b} = 0.1397$	3.1607	0.9537

Note) MLE : Maximum likelihood estimation
 MSE : Mean square error
 R^2 : Determination coefficient

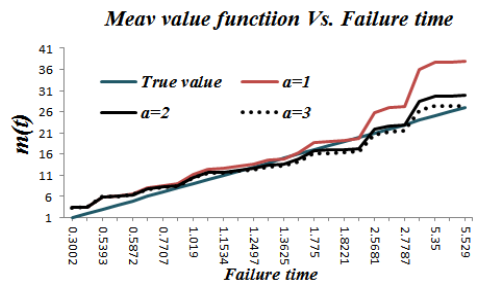
그리고 단위시간 당 고장발생률을 의미하는 위험함수는 다음과 같이 정의된다[Kim, 2019].

$$h(x) = \frac{f(x)}{1 - F(x)} = ab e^{bx} \quad (10)$$

<Table 2>의 모수추정 결과를 이용한 위험함수의 패턴은 <Figure 4>에 요약되었다. 이 그림에서 지수화-지수분포의 형상모수가 $a=1$, $a=2$ 와 $a=3$ 인 모형의 위험함수의 패턴은 비-감소 패턴을 보이고 있다. 특히 $a=1$, $a=2$ 인 경우는 $a=3$ 인 경우보다 낮게 나타나고 있으며 $a=1$ 와 $a=2$ 인 경우를 비교했을 때 $a=1$ 경우가 낮게 나타나다가 일정시간이 지나면 $a=2$ 인 경우보다 높게 나타나는 경향을 보이고 있다.



<Figure 4> Hazard Function of Each Model



<Figure 5> Mean Value Function of Each Model

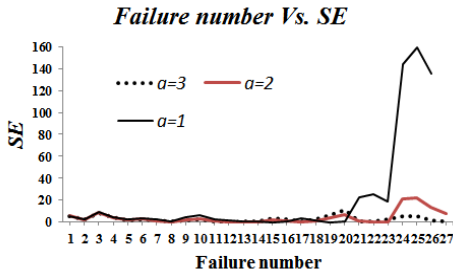
<Figure 5>에서 평균값 함수 패턴 비교에 있어서 모든 비교 모형은 비-감소형태를 이루지만 $a=1$, $a=2$ 인 경우보다 $a=3$ 인 분포가 참값과의 비교에 있어서 예측 값 추정오차가 제일 작음을 알 수 있다. 또한, 실제 값과 예측 값에 대한 차이에 대한 척도를 의미하는 평균제곱오차(MSE)의 통계량은 다음과 같다 [Kim and Kim, 2016; Kim, 2019].

$$MSE = \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{n - k} \quad (11)$$

단, $m(x_i)$ 은 관찰시점(0, x_i)까지 고장들의 누적개수이고 $\hat{m}(x_i)$ 는 x_i 시점까지 평균값함수로부터 추정된 고장의 누적개수, n 은 관찰된 고장의 수이고 k 는 추정된 모수의 수를 의미한다. <Table 2>에서 전체적인 평균 제곱오차는 $a=3$ 인 모형이 $a=1$, $a=2$ 인 모형보다 작으므로 상대적으로 정확도(Accuracy)의 척도 측면에서 효율적 모형으로 간주 할 수 있다. 구체적으로 각 고장시점에 대한 제곱오차는 식 (12)을 이용하면 다음과 유도 할 수 있다[Kim, 2019].

$$SE = \sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2 \quad (12)$$

따라서 식 (12)를 이용한 각 지점에 대한 제곱오차는 <Figure 6>에 나타내었다. 이 그림에서도 초반부 고장시점에서는 거의 유사하지만 고장시점이 증가함에 따라 $a=3$ 인 모형이 $a=1, a=2$ 인 모형보다 제곱오차 값이 작은 패턴을 보여주고 있다.



<Figure 6> Square Error of Each Failure Point

또한, 결정계수는 추정된 예측 값으로서 고장수를 설명할 수 있는 설명력을 나타내는 도구로서 정의(R^2)되고 큰 결정계수를 가진 모형은 적합성(Goodness-of-fit) 측면에서 효율적인 모형으로 간주 된다(Kim, 2017: 2019).

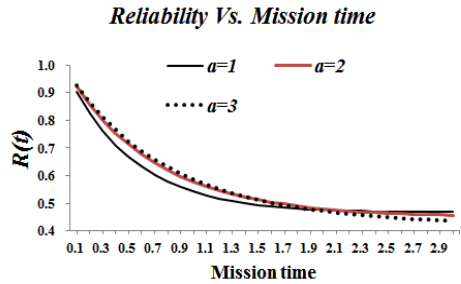
$$R^2 = 1 - \frac{\sum_{i=1}^n (m(x_i) - \hat{m}(x_i))^2}{\sum_{i=1}^n \left(m(x_i) - \frac{\sum_{j=1}^n m(x_j)}{n} \right)^2} \quad (13)$$

따라서 <Table 2>에서 결정계수 추정값도 $a=3$ 인 모형이 $a=1, a=2$ 인 모형보다 크게 나타나 적합성측면에서 효율적인 모형으로 간주 할 수 있지만 즉 제안된 모든 모형은 50% 이상을 가지며 형상모수가 클수록 상대적으로 효율적 모형으로 판단된다(Kim, 2019).

NHPP 모형에서 최종 테스트 지점 $x_{27} = 5.529$ 에서 소프트웨어 고장이 일어나고 $(5.529, 5.529 + t]$ (단, t 는 임무시간(Mission time))사이에서 소프트웨어의 고장이 일어나지 않을 확률인 조건부 신뢰도(Reliability) $\hat{R}(t | 5.529)$ 는 다음과 같은 관계로 유도된다(Kim and Kim, 2016; Kim, 2019).

$$\begin{aligned} \hat{R}(t | 5.529) &= e^{-\int_{5.529}^{5.529+t} \lambda(\tau) d\tau} \\ &= \exp[-\{m(t+5.529) - m(5.529)\}] \end{aligned} \quad (14)$$

그리고 식 (14)를 이용한 <Figure 7>의 조건부 신뢰도 함수의 형태에서 임무시간(t)이 경과함에 따라 완전히 비-증가 패턴으로 나타나고 $a=3$ 인 모형이 $a=1, a=2$ 인 모형보다 높게 나타나지만 임무시간이 경과함에 따라 낮게 나타나고 있다.



<Figure 7> Pattern of Reliability

4. 결 론

소프트웨어 개발과정에서 테스트 수행이나 실제 소프트웨어 운용과정에서 고장 발생 추이를 정량적으로 모형화하면 소프트웨어에 대한 정확성과 신뢰성을 비교 분석하여 효율성 평가를 할 수 있다.

본 연구에서는 소프트웨어 신뢰성 분야에서 널리 사용되는 지수화-지수분포와 형상모수에 따른 수명분포를 적용하고 유한고장수를 가진 비동질적인 포아송 과정을 활용한 소프트웨어의 신뢰성 속성에 대하여 비교 평가하였다. 본 연구의 결과는 다음과 같이 요약 할 수 있다. 지수화-지수분포의 형상모수에 따른 위험함수의 패턴은 비-감소 패턴을 보이고 있고 형상모수가 클수록 낮은 추세 패턴을 보이고 있다.

둘째, 평균값함수 패턴 비교에 있어서도 모든 비교 모형은 비-감소형태를 이루지만 형상모수가 클수록 참값과의 비교에 있어서 예측 값 추정오차가 제일 작기 때문에 상대적으로 정확도의 척도 측면에서 효율적 모형으로 간주 할 수 있다.

셋째, 결정계수 추정 값도 제안된 모든 모형은 50% 이상을 가지며 형상모수가 클수록 크게 나타나 적합성측면에서 형상모수가 클수록 상대적으로 효율적 모형으로 판단된다.

넷째, 신뢰도 함수의 형태는 임무시간이 경과함에 따라 완전히 비-증가 패턴으로 나타나고 있으며 형상모수가 클수록 높게 나타나지만 임무시간이 경과함에 따라 낮게 나타나고 있다.

이 연구를 통하여 소프트웨어 운용자들은 지수화-지수분포의 수명분포의 형상모수 특징을 활용한 소프트웨어 고장유형을 파악하기 위하여 평균제곱오차와 평균값 및 위험함수의 속성을 이용하면 소프트웨어 고장패턴을 파악 하는 데 기본지침으로 활용 할 수 있으리라 판단된다.

References

- [1] Goel, A. L. and Okumoto, K., "Time-dependent fault detection rate model for software and other performance measures", *IEEE Trans. Reliab.*, Vol. 28, No. 3, 1979, pp. 465-484.
- [2] Gokhale, S. S. and Trivedi, K. S., "A time/structure based software reliability model", *Annals of Software Engineering*, Vol. 8, 1999, pp. 85-121.
- [3] Kim, H.-C., "A Comparative Study on Software Reliability Models with Shape Parameter of Type-2 Gumble Life Distribution", *International Journal of Soft Computing*, Vol. 12, No. 5-6, 2017, pp. 351-354.
- [4] Kim, H.-C., "A Comparative Study on the Cost of Software Development Model Based on Inverse Exponential Distribution", *International Journal of Engineering Research and Technology*, Vol. 12, No. 11, 2019, pp. 2036-2040.
- [5] Kim, H.-C., "A Comparative Study on the Finite Failure Software Reliability Model with Modified Lindley Type Lifetime Distribution", *International Journal of Engineering Research and Technology*, Vol. 12, No. 6, 2019, pp. 760-764.
- [6] Kim, H.-C., "The Comparison Analysis about Reliability Features of Software Reliability Model Using Burr-XII and Type-2 Gumbel Lifetime Distribution", *International Journal of Engineering Research and Technology*, Vol. 12, No. 1, 2019, pp. 73-78.
- [7] Kim, K.-S. and Kim, H.-C., "The Performance Analysis of the Software Reliability NHPP Log-linear Model Depend on Viewpoint of the Learning Effects", *Indian Journal of Science and Technology*, Vol. 9, No. 37, 2016, pp. 1-5.
- [8] Oguntunde, P. E., Balogun, O. S., Okagbue, H. I., and Bishop, S. A., "The Weibull-Exponential Distribution : Its Properties and Applications", *Journal of Applied Sciences*, Vol. 15, No. 11, 2015, pp. 1305-1311.
- [9] Prasad, R. S., Rao, K. R. H., and Kantha, R. R. L., "Software reliability measuring using modified maximum likelihood estimation and SPC", *International Journal of Computer Applications*, Vol. 21, No. 7, 2011, pp. 1-5.
- [10] Song, K. Y., Chang, I. H., and Pham, H., "A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments", *Applied Science*, Vol. 7, No. 10, 2017, pp. 1-16.
- [11] Teng, X. and Pham, H., "A new methodology for predicting software reliability in the random field environments", *IEEE Transactions on Reliability*, Vol. 55, No. 3, 2006, pp. 458-468.
- [12] Yamada, S. and Osaki, S., "Software reliability growth modeling : models and applications", *IEEE Transactions on Software Engineering*, Vol. 11, No. 12, 1985, pp. 1431-1437.

■ 저자소개



김 희 철

He received a MS and a PhD from the Department of Statistics at Dongguk University in 1992 and 1998. He received a MS and a Ph.D from the Department of Statistics at Dongguk University in 1992 and 1998, respectively. He has been a professor in the Department of Industrial & Management Engineering at Namseoul University since 2005. His current research interests include software reliability engineering, computer statistics, and information systems



문 승 철

He received a MS and a PhD from the KAIST. Dept. Info. Engineering and Kookmin Univ. Dept. Info. Mgt. He is currently a Chairman for JITAM (Journal of Information Technology Application and Management) in Korea Information Technology Application and Management Society. Main interesting field : S/W Engineering, Ubiquitous computing, Datamining, Mgt. Info. Sys.