Journal of the KSQC
Vol. 19, No. 1, June. 1991

1991년 6월 韓國品質管理學會誌 제19권 제1호

SOFTWARE 信賴性 MODEL의 開發과 適用에 관한 硏究
Study on Development of Software Reliability Model and Application

金 正 子*
趙 盛 健**

ABSTRACT

According to characters of software, the methods of evaluating reliability are different.

The purpose of this study was to develop application software used in the field and to set up reliability model which failure density was used so that in the process of test, developer and user may apply a point of time to field business and to apply developed model which real data use used.

1. 序 論

1.1 硏究目的

産業現場에서 發生하고 있는 多量의 情報를 보다 迅速하고 正確하게 處理하기 위하여 우리는 Software를 情報分析의 狀況에 따라 나름대로 開發하여 情報를 處理하여 왔다.

産業現場에서 發生되는 情報를 處理하기 위해 수많은 Software가 開發되어 使用되어 왔고, 지금도 使用되고 있으며, 앞으로도 더욱 많은

* 東亞大學校 産業工學科 敎授
** 慶南專門大學 電子計算科 副敎授

Software가 開發되어 使用될 것으로 본다.

이렇게 開發되어 使用되고 있는 Software의 信賴性에 대해서는 開發者들이나 利用者들 모두가 關心을 갖지는 않았다.

그러나, Software開發 技術의 發達과 함께 Software의 開發期間과 開發環境等의 要因이 이 分野에 關心을 갖게 하여, Software의 信賴性 問題가 浮刻되게 되었다.

그래서 지금의 Software 開發者나 利用者는 막연하게 定性的으로 나타내는 信賴性이 아니라 定量的인 數値로 나타낼 수 있는 方法을 찾게 되었고, 管理者들도 定量的인 數値를 要求하게 되었다.

그래야만 開發하거나 使用하는 Software를 보다 效率的으로 管理하고 運營할 수 있게 되기 때문이다.

Software의 信賴性이란 주어진 時間동안에 要求되는 機能이 定義된 環境下에서 故障없이 遂行될 수 있는 確率을 말하며, Hardware는 이미 오래전부터 信賴性을 附與하기 위한 方法이 Model이 適用되어 다루어져 왔지만, Software는 故障을 發生시키는 메카니즘이 서로 다르기 때문에 Model을 開發하여도 實際 Software의 開發作業이나 Test作業에 適用하여 使用하기가 어려운 實情이다.

그래서 本 硏究에서는 管理와 開發을 보다 效率的으로 遂行할 수 있도록 하기 위해 實際 現場에서 많이 開發하여 使用하고 있는 Application Software를 開發하고, Test하는 過程에서 信賴性을 附與 할수 있는 時點을 찾아보기 위하여 實際 現場에서 作成한 Data를 利用해 開發된 Application Software의 Test를 행한다. Test를 遂行하면서 時間間隔에 의해 發生하는 故障의 數를 Data로 사용하여 開發된 Model에 適用해 結果를 分析해 보고자 한다.

1.2 硏究 方法

Software는 入力되는 Data에 따라서 故障이 發生 될수도 있고 發生하지 않을 수도 있다.

따라서 Software내에는 故障을 發生시킬수 있는 要因이 언제나 存在하고 있다고 볼수 있다.

그러나, 一般的으로 入力되는 Data가 Software내에 存在하는 故障 發生 要因을 건드리지 않는다면 故障은 絶對로 發生하지 않는다고 볼수 있다.

그러므로 一般的인 信賴性 Model을 開發할 때에는 觀測된 Data를 利用해 Test를 하는 過程에서 發生되는 故障의 數를 時間 函數로 하여 信賴性 Model을 開發하여 왔는데 本 硏究에서는 實行時間이 짧으면서, 故障의 頻度가 比較的 적게 나타나는 Software에 適用이 되고 있는 Logarithmic Poisson Execution Time Model을 利用하여 一般的인 Model을 開發하고자 한다.

Logarithmic Poisson Execution Time Model에서 故障密度를 구하는 一般式을 誘導하고, 구해진 式을 利用하여 實際 開發되어 業務에 適用될 Application Software에 使用할 Test Data를 만들고, Test하면서 發生하는 故障의 數로 故障密度를 구해 봄으로써 開發者나 利用者 自身이 Software의 信賴性을 附與 할수 있는 時點을 찾아 意思決定할수 있도록 한다.

2. Logarithmic Poisson Execution Time Model

一般的으로 Software의 信賴性 Model은 實行時間 t에서 觀測된 故障의 數로 나타내며,

{M(t), t≥0}으로 表現 할수 있다.

M(t)를 利用하여 平均値 函數로 表現하면 (1)式과 같이 된다.

$$\mu(t)=E\{M(t),\ t\geq 0\} \quad \cdots\cdots (1)$$

이것은 時間 t에서 期待되는 故障의 數를 말한다.

그리고, 故障密度 函數는

$\lambda(t)=\frac{d\mu(t)}{dt}$를 이용해서 誘導 할수있다.

따라서 故障 密度에 관한 平均値 函數는 (2)式과 같이 나타낼 수 있다.

$$\lambda(t)=\lambda_0 \exp[-\theta\mu(t)] \quad \cdots\cdots (2)$$

式 (2)에서 λ_0 를 初期 故障密度라 하고, θ를 故障 密度의 衰退 Parameter라고 하면 θ는 $\theta>0$이 된다.

또, 平均値 函數와 故障 密度 函數를 誘導하기 위하여 (2)式을 微分하면

$$\frac{d\mu(t)}{dt}=\lambda_0 \exp[-\theta\mu(t)] \quad \cdots\cdots (3)$$

혹은

$$\frac{d\mu(t)}{dt}=\exp[\theta\mu(t)]=\lambda, \quad \cdots\cdots (4)$$

로 나타낼 수 있다.

여기서,

$$\frac{d\ \exp[-\theta\mu(t)]}{dt}=\theta\frac{d\mu(t)}{dt}\exp[\theta\mu(t)]$$

라고 두면, 式 (4)에서

$$\frac{d}{dt}\exp[\theta\mu(t)]=\lambda_0\theta \quad \cdots\cdots (5)$$

를 얻을 수가 있다.

또, 故障當 故障密度의 減少値를 구해보면,

$$\frac{d\lambda}{d\mu}=-\lambda_0\theta\exp(\theta\mu)=-\theta\lambda \quad \cdots\cdots (6)$$

이 된다.

다시 式(5)를 積分해 보면,

$$\int \theta\mu(t)=\lambda_0\theta_1+C \text{ 가 된다.}$$

여기서 C는 積分 常數로서 $\mu(0)=0$일때, C=1이 된다.

따라서, 平均値 函數는

$$\mu(t)=\frac{1}{\theta}\ln(\lambda_0\theta_1+1) \quad \cdots\cdots (7)$$

式(7)은 t에 대한 Logarithmic 函數가 된다.

그리고, 故障의 密度 函數는

$$\lambda(t)=\frac{\lambda}{\lambda_0\theta_1+1} \quad \cdots\cdots (8)$$

이 되며, t의 逆 線型 函數가 된다.

逆 線型 函數에서 實行 時間(t)와 故障密度(λ)의 關係를 圖示하면 Fig. 1과 같이 된다.

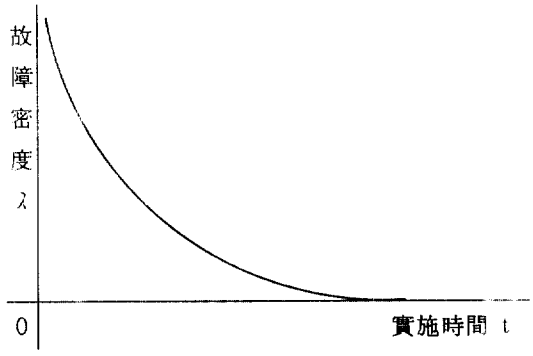

Fig. 1 Logarithmic Poisson Model에서 故障密度와 實行時間과의 關係

平均値 函數式인 (7)式과 故障密度式인 (8)式은 Software의 信賴性 Model로서 實用性과 效用性이 크기 때문에 平均値 函數나 故障 密

度 函數를 定立하는데 一般的으로 많이 使用되고 있다.

3. 信賴性 Model의 開發 段階

Software의 信賴性 Model을 開發하기 위해서는 調査된 故障 Data를 分析하여 適合한 分布의 形態를 찾고, Parameter를 利用하여 基本 Model을 作成하고, Test하여 나온 結果를 利用하여 開發者나 利用者가 意思決定을 하는 段階로 信賴性 Model을 開發하게 된다.

段階別로 細分化 해보면 다음과 같이 5段階로 나누어 볼수 있다.

1) 1段階－Software의 故障 Data의 調査段階
   이 段階에서는 故障 Data의 有效性을 調査하는 段階로서 時間間隔으로 故障의 數를 調査한다.
2) 2段階－信賴性 Model의 選擇段階
   1段階에서 調査된 Data를 分析하여 適當한 分布의 形態를 찾고 Parameter를 設定하여 Mode化 作業을 행한다.
   이 段階에서는 選擇된 Model의 說明을 容易하게 하기 위하여 必要한 假定도 設定할 수 있다.
3) 3段階－Parameter의 豫測과 Model의 適合化 段階
   選擇된 Model에 使用할 Parameter를 豫測하고, 豫測된 Parameter로 故障 Data에 대한 Model의 適合化를 행한다.
   Parameter의 豫測에는 一般的으로 最尤 推定法이 使用되지만 경우에 따라서 最小自乘法이나 그 밖의 方法들이 利用되기도 한다.
4) 4段階－Test段階
   適合化된 Model을 利用하여 Test를 하고 信賴性에 必要한 값을 구한다.
   만약 구한 豫測値에 異常이 發生하면, 追加 Data를 選擇하여 2段階의 作業으로 되돌아간다.
   그러나 얼마나 많은 Data를 追加해야 보다 適合한 Model을 만들수 있을지는 쉽게 말하기 어렵다.
5) 5段階－意思決定 段階
   이 段階에서는 구해진 信賴性의 값을 利用하여 Software에 관한 意思決定을 하게 된다.

위의 5段階를 그림으로 圖示하면 Fig. 2와 같이 된다.

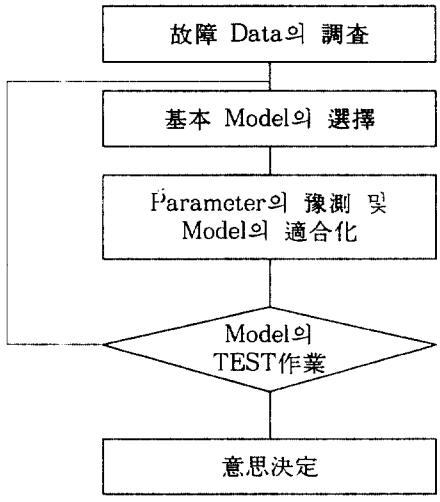

Fig. 2 Software信賴性 Model開發 段階

4. Parameter의 設定과 Model의 開發

Logarithmic Poisson Execution Time Model에서

$\lambda(t)=\lambda_0\exp\{-\theta\mu(t)\}$에 適合한 λ_0와 θ를 구하기 위하여 故障 發生의 數를 Data로 利用하여 豫測을 행한다.

여기서는 Parameter의 豫測値를 구하기 위하여 最小自乘法을 使用하며, λ_0와 θ를 $\lambda(t)$에 適合化 시키기 위해서는 Data$\{(X_i, Y_i), i=1, 2, \cdots m\}$로 豫測을 행한다.

ε_i를 Model 값에서 i번째 Data의 誤差率이라고 하면,

$\ln Y_i=\ln \lambda(X_i)+\varepsilon_i$가 된다.

따라서 $\ln Y_i=\ln\lambda_0-\theta X_i+\varepsilon_i$가 된다.

이렇게 보면, 위의 式은 Simple Linear Regression이 된다.

λ_0 와 θ를 利用하여 λ_0와 θ를 구하기 위하여 ε_i의 自乘合을 最少化한다.

$$S(\lambda_0,\theta)=\sum_{i=1}^{m}\varepsilon_i^2$$

$$=\sum_{i=1}^{m}\{\ln Y_i-\ln\lambda_0+\theta X_i\}^2$$을 最少化

하는 것이므로

最小自乘法으로 λ_0와 θ를 구하면

$$\theta=\frac{m(\sum_{i=1}^{m}\ln Y_i)-(\sum_{i=1}^{m}X_i)(\sum_{i=1}^{m}\ln Y_i)}{m(\sum_{i=1}^{m}X_i^2)-(\sum_{i=1}^{m}X_i)^2}$$

이 되며,

$$\ln\lambda o=\overline{\ln Y}-\overline{\theta X}$$

이므로 λo는 쉽게 구할수 있다.

또, 故障密度는

$\ln Y_i=\ln\lambda(X)+\varepsilon_i$

$=\ln\lambda_0-\theta X_i+\varepsilon_i$에서

$\ln Y_i=\ln\lambda(X_i)$

$=\ln\lambda_0-\theta X_i$가 되므로

$Y_i=\lambda(X_i)$

$=\lambda_0\exp(-\theta X_i)$가 된다.

따라서, 信賴性 Model은 故障密度에 관한 平均値 函數가 되므로

$\lambda(t)=\lambda_0\exp\{-\theta\mu(t)\}$가 된다.

5. Model을 利用한 Data分析

觀測된 故障發生의 Data(X_i, Y_i)는 Table 1.과 같다.

No	X(時間)	Y(故障數)	故障數 累積値
1	1	14	14
2	2	16	30
3	3	12	42
4	4	10	52
5	5	11	63
•	•	•	•
•	•	•	•
12	12	2	93
13	13	3	96
14	14	1	97
15	15	2	99

Table 1. 觀測된 故障 Data와 累積値

구해진 信賴性 Model을 適合化 시키기 위하여 最小自乘法으로 λ_0 와 θ를 구해보면,

$\lambda_0=29.207$이 되고, $\theta=0.282$가 되므로

故障密度에 대한 平均値 函數는

$\lambda(t)=29.207\exp(-0.282t)$가 된다.

故障密度에 관한 平均値函數를 利用하여 各 時點에서의 故障密度를 구해 보면 Table 2.와 같다.

t	1	5	7	9	11	14	18	20	24	26	27	28	29
λ	22.03	7.13	4.06	2.31	1.31	0.56	0.18	0.10	0.33	0.02	0.01	0.01	0.00

Table 2. 各 視點에서의 故障密度

따라서 開發되는 Software는 Table. 2에서 볼수있듯이 故障密度가 0.1以下인 때가 t=20인 時點이 되고, 0.0이 되는 때가 t=29인 時點임을 알수 있다.

이것은 단지 測定된 故障의 數를 使用해 故障密度에 관한 平均値 函數로 구한 結果이므로 구해진 값을 利用하여 Model 作成段階의 마지막 過程인 Software自體에 대한 意思決定은 오직 Software를 開發한 開發者나 利用할 利用者가 適當한 時點을 찾아 意思決定을 해야한다.

6. 結　　論

Software의 信賴性 評價 Model은 Software의 特性에 따라서 差異가 많이 發生하며, 특히 開發者의 能力이나 開發環境 等이 Software의 特性을 決定하는데 있어서 아주 重要한 要因이 될 것이다.

따라서 Software를 評價하는 Model의 開發에는 반드시 開發者의 能力이나 環境要因 等을 變數로 하는 信賴性 Model이 開發되어야 하는데 여기에는 定性的인 Data의 定量化 作業이 優先되어야 하기 때문에 어려움이 많이 따르게 된다.

結局 이런 變數들은 開發된 Software에 障碍要因으로 潛在하게 되고, 結果的으로는 Test過程에서 障碍로 나타나게 된다고 볼수 있다.

그래서, 本 硏究에서는 위에서 言及한 여러가지의 變數가 Test過程에서 障碍로 導出되었다고 보고, 開發된 Sftware를 利用하여 一定時點(0, t_i)까지의 故障 Data를 調査하여 各各의 時點에서 發生하는 故障의 密度를 구해 보므로서 Software에 信賴性을 附與 할수 있는 時點을 찾아 開發者나 利用者가 意思決定을 할수 있게 했다.

여기서 알수 있는것은 Logarithmic Poisson Execution Time Model은 Software의 Test時間이나 開發期間이 짧게 要求되는 경우에 適當한 方法으로서 實際 産業現場에서 適用하기가 좋은 Software 信賴性 評價 方法이다.

本 論文에서 使用한 Data는 現場에서 使用하게 될 Application Software를 利用하였고, 開發된 信賴性 Model이 使用하기가 簡便하기 때문에 實際 現場에서 開發하거나 使用하고 있는 Application Software에 適用이 可能한 Moldel이라고 生覺한다.

本 硏究에서는 故障 密度를 利用하여 Software의 信賴性을 附與하였지만 故障密度가 아닌 確率로 나타내 주었다면 더 좋았을 것이며, 殘存 故障의 數를 各各의 時點에서 구해 주었으면 하는 아쉬움이 남는다.

참 고 문 헌

1. 朴聖炫.,(1981), "回歸分析", 大英社
2. 鈴木和幸 外1.,(1984), "品質保證と 信賴性", 日科技連
3. 佐和隆光 外1.,(1980), "回歸分析の 實際", 新曜社
4. みつはし たけし.,(1978), "質的 データ解析法", 日科技連
5. いちだ たかい 外1.,(1990), "信賴性の 分布と 統計", 日科技連
6. BEV LITTEWOOD.,(1980), "theoris of Software Reliability: How Good Are They and How Can They Be Improved?", IEEE SOFTWARE ENGINNERING
7. GEORGE J. SCHIK.,(1987), "An Analysis of Competing Software Reliability Models", IEEE SOFTWARE ENGINEERING
8. IAM SOMMERVILLE.,(1984), "Software Engineering", ADDISON WESLEY
9. JOHND, MVSA .,(1975), "A Theory of software Reliability and Its Application", IEEE SOFWARE ENGINEERING.
10. JOHN D, MUSA.,(1975), "Software Reliability", McGRAW-HILL
11. M. L. JAMES.,(1985), "Applied Numerical Methods for Digital Computation", Harpper and Row
12. Robert N. Charette.,(1988), "Software Engineering Environments : Concepts and Technology", McGRAW-HILL INTERNATIONAL EDITIONS
13. SHELDON M. ROSS.,(1984), "Introduction to Probability Models", ACADEMIC PRESS
14. SHIGERU YAMADA.,(1985), "Software Reliability Growth Modeling: Models and Applications", IEEE SOFTWARE ENGINEERING.