B-008

組込みシステムのオープンソースソフトウェア移植工程に対する信頼性評価のための一般化ハザードレートモデルと適合性評価

Generalized Hazard Rate Modeling with Goodness-of-Fit Comparison for the Open Source Software Porting-Phase of Embedded Systems

吉田 祐貴* 田村 慶信† 山田 茂‡
Yuki Yoshida Yoshinobu Tamura Shigeru Yamada

1 はじめに

オープンソースソフトウェア（open source software，以下 OSS と略す）は，世界中の誰もが開発に参加でき，誰でも自由に改変可能なソフトウェアである．そのため，最近では組込みシステムやサーバ用途として広く採用され，急激に普及が広まっている [1]，[2]．また，電子行政機関がオープン規格や OSS を利用することで，プライバシーや個人の自由を保護し，市民と電子政府の間で情報のやり取りを可能にするために役立つことから，EU 加盟国を中心に欧米においても政府関係機関が OSS を支持する動きが広がっている [3]．

特に，OSS の最近の傾向として，組込み機器に対しても Android[4] や BusyBox[5] などに代表される組込み OSS が積極的に採用されている．中でも Android は，スマートフォン等の携帯端末のプラットフォームとして組込み OSS の中でも多くの注目を集めている．現在，Android を搭載したスマートフォンは，HTC や Samsung，Motorola などの企業により開発・発表されており，既に 20 機種以上が市場に出荷されている．このような現状から，今後組込みシステム開発における Android などの組込み OSS の利用は，益々拡大していくものと思われる．

一方で，OSS の利用に関しては，OSS の普及を妨げる大きな要因として，サポートや品質上の問題が指摘されている [3]．OSS の開発環境を考えた場合，ユーザの使用により不具合が確認されるとバグトラッキングシステム上に不具合内容が報告され，その内容に基づきソースコードの修正作業を開発者が行い，修正された OSS を再度，公表・配布するという開発サイクルで成り立っている．このように，OSS はウォーターフォールモデルに代表されるような一般的なソフトウェア開発モデルの下で開発が進められていないため，開発工程には特に定められたテスト工程が存在せず，また，開発から運用・保守に及ぶ工程においてソフトウェアの品質・信頼性を評価するという試みも行われていない．

オープンソース・プロジェクトのメンバー構成を考えた場合，開発ボランティアは，中心的なソフトウェアを担当するコア開発者と，中心部分に付随するソフトウェアを担当する周辺開発者に大別される．プロジェクトにおける最重要メンバーは，中心的なソフトウェアを担当するコア開発者であり，彼らは中心的なメーリングリストにアクセス可能となっている．そのため，この指導的集団を形成している主要メンバーが主導的にテスト進捗度管理技術を導入することにより，より高品質な OSS の開発に結びつくものと考える．

しかしながら，OSS に対する現在の研究動向としては，設計技法や開発手法，セキュリティを対象とした文献はいくつか提案されているものの [6]-[9]，動的解析に基づいた組込み OSS に対する信頼性評価に関する研究はほとんど行われていないのが現状である．さらに，OSS の信頼性評価に関する特徴として，サーバおよびアプリケーションソフトウェアについては信頼度成長曲線に関して一定の傾向を示すものが多いが [10]，[11]，組込みソフトウェアについては，ハードウェアに依存するコンポーネントが含まれていることから，信頼性を評価することが難しい．

従来から，ソフトウェア製品の開発プロセスにおけるテスト進捗管理や出荷品質の把握のための信頼性評価を行うアプローチとして，ソフトウェア故障の発生現象を不確定事象として捉えて確率・統計論的に取り扱う方法がとられている．その代表的かつ古典的モデルの 1 つとして，ハザードレートモデルがある [12]-[16]．

本論文では，オープンソースプロジェクトの下で開発されている組込み OSS を採用する際の移植作業（ポーティング）工程に対する信頼性評価法を提案する．特に，企業組織において独自に開発された基板上へ組込み OSS を導入する際においては，独自に開発されたデバイスドライバと組込み OSS との整合性を確認する作業も重要となる．本論文では，こうしたソフトウェアコンポーネントと組込み OSS を同時に考慮したハザードレートモデルを提案する．また，OSS のバグトラッキングシステム上に登録されたフォールトデータに基づく信頼性評価例を示すとともに，予測相対誤差と平均偏差 2 乗和を適合性評価尺度とした適合性比較を行うことで，実際に公開されている組込み OSS を採用した移植作業工程における信頼性評価法の適用可能性について考察する．

これにより，組込み OSS の普及を妨げる大きな要因として考えられている品質上の問題に対して，信頼性という観点からなんらかの定量的指標を提示することが可能となるものと考える．

2 OSS 移植工程に対する信頼性評価法

2.1 移植工程のためのハザードレートモデル

本論文では，組込み OSS のポーティング時における動的実行環境，すなわち独自に開発されたハードウェアに対する組込み OSS の移植作業中に生じるソフトウェア故障には，次の 2 種類があるものと仮定する．

*山口大学大学院理工学研究科電子情報システム工学専攻
†山口大学大学院理工学研究科
‡鳥取大学大学院工学研究科

A1. 組込み OSS に潜在するフォールトにより引き起こされるソフトウェア故障.

A2. 独自に開発されたソフトウェアコンポーネントに内在するフォールトにより引き起こされるソフトウェア故障.

ここで，ソフトウェア故障とは，ソフトウェア内に潜在するフォールト（バグ）により期待通りに動作しないことと定義する [12]. また，1 つのソフトウェア故障は 1 個のフォールトにより引き起こされると仮定し，発生したソフトウェア故障の原因となるフォールトは A1 と A2 のどちらによるものか区別できないものとする．ここで，A1 のソフトウェア故障は確率 p で発生し，A2 のソフトウェア故障は確率 $(1-p)$ で発生するものとする．このとき，$(k-1)$ 番目と k 番目のソフトウェア故障の時間間隔を確率変数 X_k で表すものとすると，X_k に対するハザードレート関数 $z_k(x)$ は，

$$z_k(x) = p\cdot z_k^1(x) + (1-p)\cdot z_k^2(x) \quad (k=1,2,\cdots; 0\le p\le 1), \tag{1}$$

$$z_k^1(x) = D(1-\alpha\cdot e^{-\alpha k})^{k-1} \quad (k=1,2,\cdots; -1<\alpha<1,\ D>0), \tag{2}$$

$$z_k^2(x) = \phi\{N-(k-1)\} \quad (k=1,2,\cdots, N; N>0, \phi>0), \tag{3}$$

により表すことができるものと仮定する．ここで，各諸量を次のように定義する．

- $z_k^1(x)$： A1 に対するハザードレート,
- α： OSS の活動状態を表す形状パラメータ,
- D： 1 番目のソフトウェア故障に対する初期ハザードレート,
- $z_k^2(x)$： A2 に対するハザードレート,
- N： 独自に作られたソフトウェアコンポーネント内に潜在する総固有フォールト数,
- ϕ： ソフトウェアコンポーネント内の残存フォールト 1 個当りのハザードレート,
- p： $z_k^1(x)$ に対する重みパラメータ.

式 (1) は，組込み OSS 内に潜在する総固有フォールトおよび独自に開発されたコンポーネントに内在するフォールトによるソフトウェア故障発生現象を，発生割合を表す p および $(1-p)$ により陽に記述するものである．本モデルに含まれる式 (2) は，既存の Moranda モデル [14] を組込み OSS の開発環境に合わせて修正したものであり，組込み OSS のソフトウェア故障発生事象を表すハザードレートを意味する．また，式 (3) は既存の Jelinski-Moranda(J-M) モデル [13] であり，独自のソフトウェアコンポーネントのソフトウェア故障発生事象を表すハザードレートを意味する．特に，式 (2) は，1 番目のソフトウェア故障に対する初期ハザードレートが OSS の活動状況に応じて幾何級数的に減少するとともに，OSS の活動状態が指数関数的に増加するものと仮定している．

2.2 信頼性評価尺度

組込み OSS のポーティング時に検出される $(k-1)$ 番目と k 番目の間のソフトウェア故障の発生間隔を表す確率変数 X_k の分布関数は，

$$F_k(x)\equiv \mathrm{P_r}\{X_k\le x\} \quad (x\ge 0), \tag{4}$$

により定義され，時間区間 $(0, x]$ でソフトウェア故障が発生する確率を表す．ここで，確率 $\mathrm{P_r}\{A\}$ は事象 A が発生する確率を意味する．したがって，$F_k(x)$ の導関数

$$f_k(x)\equiv \frac{dF_k(x)}{dx}, \tag{5}$$

は，X_k の確率密度関数である．また，時間区間 $(0, x]$ でソフトウェア故障が発生しない確率を意味するソフトウェア信頼度は，

$$R_k(x)\equiv \mathrm{P_r}\{X_k > x\} = 1 - F_k(x), \tag{6}$$

により定義される．式 (4) および式 (5) から，時間区間 $(0, x]$ でソフトウェア故障が発生してないときに，引き続く単位時間内にソフトウェア故障が発生する割合を意味する故障率（ハザードレート）を

$$z_k(x)\equiv \frac{f_k(x)}{1-F_k(x)} = \frac{f_k(x)}{R_k(x)}, \tag{7}$$

により与えることができる．したがって，式 (1) のハザードレートモデルから，種々の信頼性評価尺度を導出できる．確率密度関数は，

$$f_k(x) = \{pD(1-\alpha\cdot e^{-\alpha k})^{k-1} + (1-p)\phi(N-k+1)\} \cdot \exp\left[-\{pD(1-\alpha\cdot e^{-\alpha k})^{k-1} + (1-p)\phi(N-k+1)\}\cdot x\right], \tag{8}$$

となる．また，ソフトウェア信頼度は，

$$R_k(x) = \exp\left[-\{pD(1-\alpha\cdot e^{-\alpha k})^{k-1} + (1-p)\phi(N-k+1)\}\cdot x\right], \tag{9}$$

と表すことができる．さらに式 (6) より，X_k の平均値すなわち k 番目のソフトウェア故障に対する平均ソフトウェア故障時間間隔（Mean Time between Software Failures, 以下 MTBF と略す）は，

$$\mathrm{E}[X_k] = 1\Big/\left\{pD(1-\alpha\cdot e^{-\alpha k})^{k-1} + (1-p)\phi(N-k+1)\right\}, \tag{10}$$

により与えられる．

3 複数のコンポーネントを考慮した信頼性評価法

3.1 一般化ハザードレートモデル

前述したハザードレートモデルはコンポーネントが 1 つの場合を想定したものである．ここでは，複数のコンポーネントについて考慮した一般化ハザードレートモデルを提案する．

ソフトウェア故障の発生事象，種類およびそれらに付随する仮定は，先に定義したハザードレートモデルと同一のものであり，確率 p_0 で A1 を，確率 p_i で A2 のソフトウェア故障が発生するものとする．このとき，$(k-1)$ 番目と k 番目のソフトウェア故障の時間間隔を確率変数 X_k とすると，X_k に対するハザードレート関数 $z_k(x)$ を，

$$z_k(x) = p_0 \cdot z_k^0(x) + \sum_{i=1}^{m} p_i \cdot z_k^i(x) \quad (11)$$
$$(k = 1, 2, \cdots; 0 \leq p \leq 1),$$
$$z_k^0(x) = D(1 - \alpha \cdot e^{-\alpha k})^{k-1} \quad (12)$$
$$(k = 1, 2, \cdots; -1 < \alpha < 1, D > 0),$$
$$z_k^i(x) = \phi_i\{N_i - (k-1)\} \quad (13)$$
$$(i = 1, 2, \cdots, m, k = 1, 2, \cdots, N_i$$
$$, N_i > 0, \phi_i > 0),$$

のように表すことができるものと仮定する．ここで，各諸量を次のように定義する．

- $z_k^0(x)$： A1 に対するハザードレート，
- α： OSS の活動状態を表す形状パラメータ，
- D： 1 番目のソフトウェア故障に対する初期ハザードレート，
- p_0： 組込みソフトウェア全体に対する組込み OSS の開発労力の割合，
- $z_k^i(x)$： A2 に対する i 番目のコンポーネントに対するハザードレート，
- m： コンポーネント数，
- N_i： i 番目のコンポーネント内に潜在する総固有フォールト数，
- ϕ_i： i 番目のコンポーネントに対する固有フォールト 1 個当りのハザードレート，
- p_i： 組込みソフトウェア全体に占める i 番目のコンポーネントの開発労力の割合．

式 (11) は，組込み OSS 内に潜在する総固有フォールトおよび独自に開発された i 番目のコンポーネントに内在するフォールトによるソフトウェア故障発生現象を，発生割合を表すパラメータ p_0 および p_i により陽に記述するものである．p_0 および p_i には，ソースコード行数，開発コスト，開発時間に関する割合などを適用することができる．

3.2 信頼性評価尺度

式 (11) のハザードレートモデルから，信頼性評価尺度を導出できる．まず確率密度関数は，

$$f_k(x) = \{pD(1 - \alpha \cdot e^{-\alpha k})^{k-1} + \sum_{i=1}^{m} p_i \cdot \phi_i(N_i - k + 1)\}$$
$$\cdot \exp[-\{pD(1 - \alpha \cdot e^{-\alpha k})^{k-1}$$
$$+ \sum_{i=1}^{m} p_i \cdot \phi_i(N_i - k + 1)\} \cdot x], \quad (14)$$

となる．また，ソフトウェア信頼度は，

$$R_k(x) = \exp[-\{pD(1 - \alpha \cdot e^{-\alpha k})^{k-1}$$
$$+ \sum_{i=1}^{m} p_i \cdot \phi_i(N_i - k + 1)\} \cdot x], \quad (15)$$

と表すことができる．さらに，X_k の平均値すなわち k 番目の MTBF は，

$$\mathrm{E}[X_k] = 1 \Big/ \Big\{ pD(1 - \alpha \cdot e^{-\alpha k})^{k-1}$$
$$+ \sum_{i=1}^{m} p_i \cdot \phi_i(N_i - k + 1) \Big\}, \quad (16)$$

により与えられる．

4 適合性評価

本論文では，モデルの適合性評価尺度として，各モデルの予測値に対する予測相対誤差と平均偏差 2 乗和を用いる．

まず予測相対誤差 (predicted relative error) は，任意の時刻 t_k において推定したときの，フォールト報告終了時点 t_K までに発見される累積フォールト数とデータの予測値との相対誤差である．ある任意の時点 t_k における予測相対誤差を $PRE_k[t_k]$ とすると，

$$PRE_k[t_k] = \frac{\hat{y}(t_k; t_K) - y_K}{y_K}, \quad (17)$$

により計算される．$\hat{y}(t_k; t_K)$ は任意の時点 t_k までの実測データを用いたフォールト報告終了時点 t_K でのソフトウェア故障発生時間間隔の推定値，y_K はフォールト報告終了時点 t_K までに発見されたソフトウェア故障発見時間間隔の実測値である．

また平均偏差 2 乗和 (mean square error) は，実測値と推定値との 2 乗誤差をデータ数で平均化したものである．一定のテスト時点 t_k までに発生したソフトウェア故障発生時間間隔 y_k に関する K 組のソフトウェア故障発生時間間隔データ $(t_k, y_k)(k = 1, 2, \cdots, K)$ が観測されているものとすると，平均偏差 2 乗和 (MSE) は，

$$\mathrm{MSE} = \frac{1}{K} \sum_{k=1}^{K} (y_k - \hat{y}_k)^2, \quad (18)$$

により与えられる．ここで，$\hat{y}_k$ は任意の時点 $t_k (k = 1, 2, \cdots, K)$ における推定値を表す．

5 数値例

5.1 組込み OSS

本論文では，携帯電話用 OS として開発・公開されている Android[4] 上で BusyBox[5] が動作するシステムを構築する環境を想定し，Android が A1 に対するソフトウェア故障を，BusyBox が A2 に対するソフトウェア故障を表すものと仮定する．移植作業工程を想定するために，実際の Android および BusyBox のオープンソースプロジェクトにおけるバグトラッキングシステム上に登録されたフォールトデータを適用した数値例を示す．Android は携帯電話用 OS として知られ，BusyBox はテレビ，オーディオ，ブロードバンドルータ，小型サーバなど，家電製品を代表とした様々な組込み製品に利用されている．本論文で

は，Android 1.5 NDK，Release 1 以降のデータを採用し，BusyBox については， BusyBox 1.10.1 (stable) 以降のデータを適用した数値例を示す．

5.2 モデルパラメータの推定

数値例を示すにあたって，各モデルパラメータの推定には非線形最小二乗法である Levenberg-Marquardt 法を用いた．ここで，OSS 移植工程に対するハザードレートモデルのパラメータ推定に際しては，簡単化の為に $pD = w1$, $(1-p)\phi = w2$ として推定を行った．

また，一般化ハザードレートモデルのパラメータ推定に際して，p_0 および p_i については，これまでに発見された累積発見フォールト数の割合を適用した．まず，Android については，Android Market introduced から Android 1.5 NDK，Release 1 までの累積発見フォールト数は 277 個であり，BusyBox に関しては， BusyBox 1.10.0 (unstable) から BusyBox 1.10.1 (stable) までの累積発見フォールト数は 21 個であった．本論文では，BusyBox の主要コンポーネントを buildroot および BusyBox と仮定し，その他の uClibc のようなコンポーネントをサブコンポーネントと仮定した場合を想定する．この場合，コンポーネント数は $m = 2$ となる．したがって，パラメータ p_0, p_1 および p_2 は以下のように与えられる．

$$\widehat{p_0} = 0.92953,\ \widehat{p_1} = 0.06040,\ \widehat{p_2} = 0.01007.$$

上述のように与えられた $\widehat{p_0}$, $\widehat{p_1}$, および $\widehat{p_2}$ から，モデルに含まれる未知パラメータ $\widehat{D}, \widehat{\alpha}, \widehat{\phi_i}, \widehat{\phi_2}, \widehat{N_1}$, および $\widehat{N_2}$ を推定した．

5.3 評価結果

5.3.1 ハザードレートモデル

まず，ソフトウェア信頼度の推定値 $\widehat{R_{30}}(x)$ を図 1 に示す．図 1 より，OSS 移植工程に対するハザードレートモデルは 0.075 日経過後に他のハザードレートモデルよりも楽観的な推定結果になることが分かる．

次に，推定された MTBF を図 2 に示す．図 2 より，OSS 移植工程に対するハザードレートモデルは他のハザードレートモデルよりも適合性が高いことが確認できる．

5.3.2 一般化ハザードレートモデル

まず，ソフトウェア信頼度の推定値 $\widehat{R_{30}}(x)$ を図 3 に示す．図 3 より，一般化ハザードレートモデルは 0.05 日経過後に他のハザードレートモデルよりも楽観的な推定結果になることが分かる．

次に，推定された MTBF を図 4 に示す．図 4 より，一般化ハザードレートモデルの推定 MTBF は J-M モデルに酷似していることが分かる．また，図 2 のハザードレートモデルと比較すると，MTBF については一般化ハザードレートモデルの方が楽観的に見積もっている様子が確認できる．

5.3.3 適合性評価結果

まず，OSS 移植工程に対するハザードレートモデルの予測相対誤差を図 5 に示す．図 5 より，OSS 移植工程対する

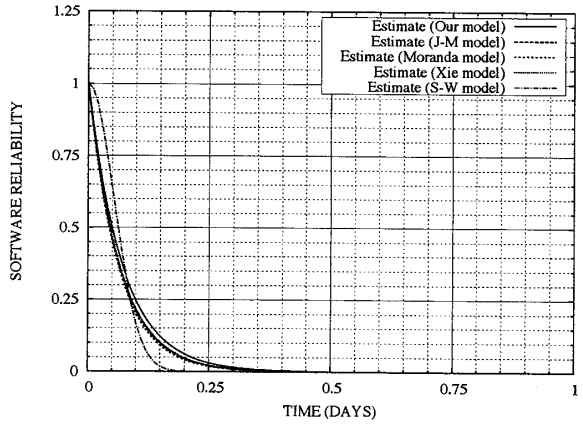

図 1： ハザードレートモデルに対する推定されたソフトウェア信頼度．

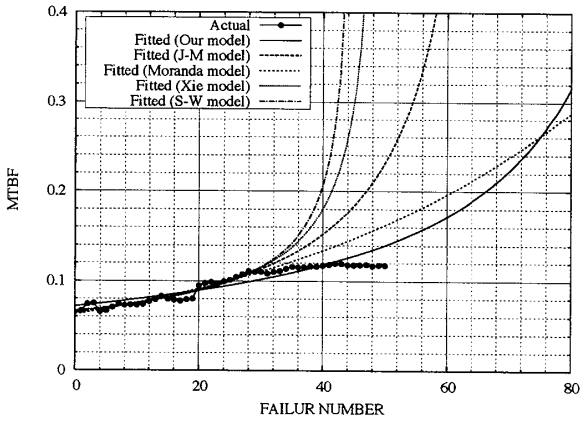

図 2： ハザードレートモデルに対する推定された MTBF．

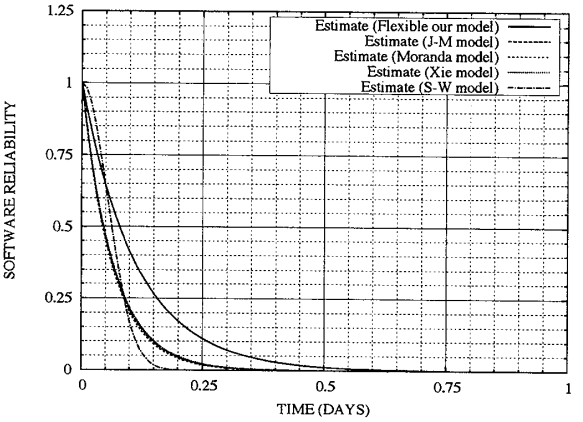

図 3： 一般化ハザードレートモデルに対する推定されたソフトウェア信頼度．

ハザードレートモデルは，フォールトデータ数の 60 %以降で他のハザードレートモデルよりも予測相対誤差が小さくなっていることが分かる．

次に，一般化ハザードレートモデルの予測相対誤差を図 6 に示す．図 6 より，一般化ハザードレートモデルの予測相対誤差は，推定 MTBF と同様に J-M モデルと酷似しており，Moranda モデルの方が適合性が良好である様子が確認できる．これは，フォールト報告終了時点までのデータに基づいて推定されたパラメータ p_0, p_1, および p_2 の値を固定したままで推定した結果であるため，フォールト報告終了時点以降の挙動については不確実性が高く，比較対象として妥当ではないと考える．

さらに，推定された各モデルの平均偏差 2 乗和を表 1 に示す．表 1 より，OSS 移植工程に対するハザードレートモデルの平均偏差 2 乗和は，30 日までのデータに対しては他のハザードレートモデルと大差ないが，50 日までのデータに対してはいずれのハザードレートモデルよりも値が小さく，適合性が良好であることが確認できる．

上述した結果から，OSS 移植工程に対するハザードレートモデルは，組込みシステムの OSS 移植工程に対する信頼性評価において適合性が高いことが分かる．

表 1： 推定された平均偏差 2 乗和．

	MSE(30 days)	MSE(50 days)
Our model	4.8467×10^{-5}	6.2665×10^{-5}
Flexible our model	1.9545×10^{-5}	1.0091×10^{-3}
J-M model	1.9545×10^{-5}	1.0102×10^{-3}
Moranda model	2.2923×10^{-5}	2.0228×10^{-4}
Xie model	7.3864×10^{-3}	4.1080×10^{-2}
S-W model	1.9268×10^{-5}	7.1464×10^{-3}

6 おわりに

本論文では，オープンソースプロジェクトの下で分散共同開発されている組込み OSS を採用した組込みシステムの移植作業工程に対する信頼性評価法を提案した．また，実際の OSS のバグトラッキングシステムに登録されているフォールトデータに対して，信頼性評価尺度に関する数値例を示した．

組込み OSS を採用した組込みシステム開発においては，移植作業の成否が組込み製品を出荷できるどうかに直接的に関係することから，組込みシステムの開発工程の中でも移植工程を適切に管理することは非常に重要となる．特に，組込み OSS のソフトウェア故障発生時間間隔データに関しては，ソフトウェア故障発生数に比例して MTBF が増加する傾向があるものとそうでないものとが存在するため，それに応じた適切なハザードレートモデルを選択する必要がある．本論文では，組込み OSS に対するハザードレートモデルを構築するとともに，組込みシステムを構成するデバイスドライバのような複数のコンポーネントを同時に考慮したハザードレートモデルを提案した．また，実際の移植作業工程を想定した数値例を示すとともに，予

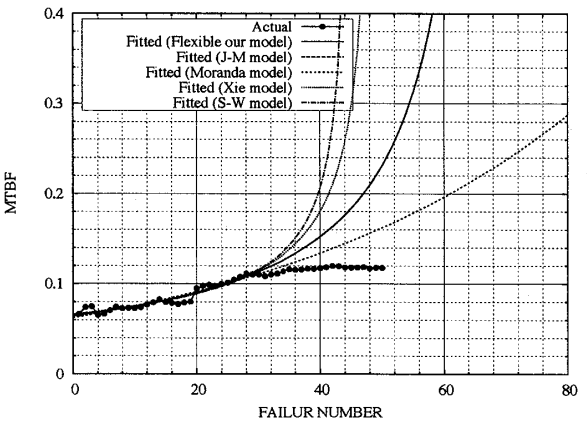

図 4： 一般化ハザードレートモデルに対する推定された MTBF．

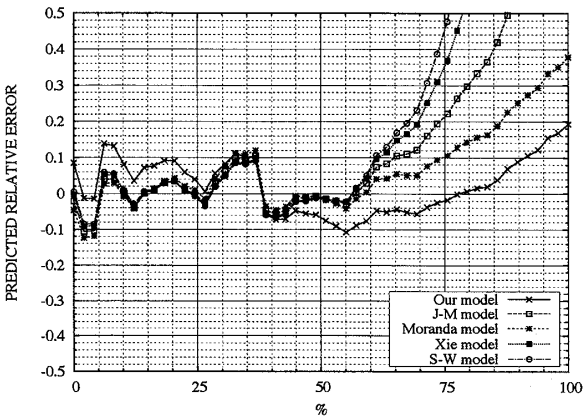

図 5： 推定されたハザードレートモデルに対する予測相対誤差の比較結果．

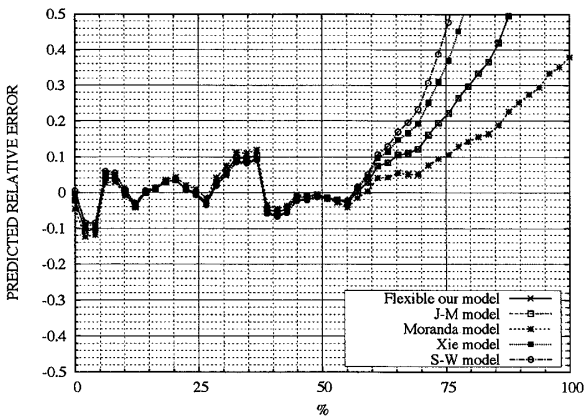

図 6： 推定された一般化ハザードレートモデルに対する予測相対誤差の比較結果．

測相対誤差と平均偏差２乗和を評価尺度として適合性評価を行った．

組込み OSS の普及の流れを阻害する要因として，サポートや品質上の問題が挙げられる．本論文では，このような問題を解決するためにオープンソースプロジェクトの下で開発された組込み OSS の移植作業工程に対する信頼性評価法の１例を示した．本論文の数値例で取り上げた Android および BusyBox は，機器のネットワーク化，開発コスト削減，オープンソースといった点から組込み OS として近年注目されている．今後もオープンソースプロジェクトに基づく開発形態は急速に発展するものと考えられることから，こうした組込み OSS の信頼性および移植性評価法として利用できるものと考える．

組込みシステム開発の移植作業工程において，ある程度目安となるような適切な移植作業期間を推定することは，リリース後の信頼性維持や進捗度管理に役立つと考えられる．これまでにも，一般的な企業組織において開発されたソフトウェアシステムを対象としたソフトウェアの最適リリース問題が数多く提案されている．組込み OSS を利用した組込みシステム開発においても，移植作業工程における最適リリース問題として総期待ソフトウェアコストを定式化することにより，最適な移植作業期間を決定することができるものと考える．将来的には，上述したような最適リリース問題を定義し，移植作業工程における最適リリース時刻を推定する必要があるものと考える．

謝辞

本研究の一部は，文部科学省科学研究費基盤研究 (C)（課題番号 22510150）および若手研究 (B)（課題番号 21700044）の援助を受けたことを付記する．

参考文献

[1] The Apache HTTP Server Project, Apache Software Foundation, http://httpd.apache.org/

[2] Mozilla.org, Mozilla Foundation, http://www.mozilla.org/

[3] ソフトウェア情報センター研究会報告書, オープンソースソフトウエアの利用状況調査／導入検討ガイドラインの公表について, 東京, 2004.

[4] Open Handset Alliance, Android, http://www.android.com/

[5] Eric Andersen, BusyBox, http://www.busybox.net/

[6] A. MacCormack, J. Rusnak, and C.Y. Baldwin, "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," Informs J. Management Science, vol. 52, no. 7, pp.1015–1030, 2006.

[7] Y. Zhoum, J. Davis, "Open source software reliability model: an empirical approach," Proc. the Workshop on Open Source Software Engineering (WOSSE), vol. 30, no. 4, 2005, pp. 67–72.

[8] P. Li, M. Shaw, J. Herbsleb, B. Ray and P. Santhanam, "Empirical evaluation of defect projection models for widely-deployed production software systems," Proc. the 12th Intern. Symp. the Foundations of Software Engineering (FSE-12), 2004, pp. 263–272.

[9] J. Norris, "Mission-critical development with open source software," IEEE Software, vol. 21, no. 1, 2004, pp. 42–49.

[10] Y. Tamura and S. Yamada, "Software reliability assessment and optimal version-upgrade problem for open source software," Proc. of the 2007 IEEE Intern. Conf. on Systems, Man, and Cybernetics, Montreal, Canada, October 7.10, pp. 1333–1338, 2007.

[11] Y. Tamura and S. Yamada, "A method of user-oriented reliability assessment for open source software and its applications," Proc. of the 2006 IEEE Intern. Conf. on Systems, Man, and Cybernetics, Taipei, Taiwan, Oct. 8.11, pp. 2185–2190, 2006.

[12] 山田茂, ソフトウェア信頼性モデル－基礎と応用－, 日科技連出版社, 東京, 1994.

[13] Z. Jelinski, P.B. Moranda, Software Reliability Research, in Statistical Computer Performance Evaluation, Freiberger, W.(ed.), pp. 465–484, Academic Press, New York, 1972.

[14] P.B. Moranda, "Event–altered Rate Models for General Reliability Analysis," IEEE Trans. Reliability, vol. R-28, no. 5, pp. 376–381, 1979.

[15] M. Xie, "On a Generalization of the J-M Model," Proc. Reliability'89, p. 5, Ba/3/1–5 Ba/3/7, 1989.

[16] G.J. Schick and R.W. Wolverton, "An Analysis of Competing Software Reliability Models," IEEE Trans. Reliability Engineering, vol. SE-4, no. 2, pp. 104–120, 1978.