

機関番号：32675  
 研究種目：基盤研究（C）  
 研究期間：2008～2010  
 課題番号：20500036  
 研究課題名（和文）ソフトウェアの信頼性評価シミュレータとそのキャリブレーション法の開発  
 研究課題名（英文）Development of Software Reliability Assessment Simulator and Its Calibration Method  
 研究代表者 木村 光宏（KIMURA MITSUHIRO）  
 法政大学・理工学部・教授  
 研究者番号：20263486

研究成果の概要（和文）：我々は大規模なソフトウェアのための精度のよい信頼性評価手法の一つとして、過去に簡単なシミュレーションモデルによる方法を提案した。これを実際のソフトウェア信頼性の評価に用いるためには、シミュレーターに含まれるパラメータを、実際のテストデータに平均的に合致するよう較正（キャリブレーション）することが必須である。この研究では、そのキャリブレーションの方法を開発した上で、実際に得られているソフトウェアテストに関するデータの解析を通じて新しいソフトウェアの信頼性評価法を提案し、この手法の有効性を示した。

研究成果の概要（英文）：In order to measure the degree of reliability of large scale software systems, many software reliability assessment models have been proposed by many researchers so far. In such a situation, we have developed several simulation models to assess software reliability based on the time series data. Especially, we provide a calibration scheme of control parameters included in the software reliability simulator. The control parameters are calibrated so as to minimize the mean sum of squared errors between the actually observed data and the generated ones by the simulator. We showed several numerical examples of the calibration and software reliability assessment by the proposed scheme. We confirmed that these results met our aim.

#### 交付決定額

（金額単位：円）

	直接経費	間接経費	合計
2008年度	1,500,000	450,000	1,950,000
2009年度	1,000,000	300,000	1,300,000
2010年度	900,000	270,000	1,170,000
総計	3,400,000	1,020,000	4,420,000

研究分野：ソフトウェア信頼性工学

科研費の分科・細目：情報学・ソフトウェア

キーワード：ソフトウェア信頼性、シミュレーション、ソフトウェア開発管理

#### 1. 研究開始当初の背景

1970年代以降、数多くの研究者らによって、ソフトウェアの信頼性を定量的に評価するための枠組みとしての確率モデル、特に大規模ソフトウェア開発の最終工程として設けられた総合テストなどの際に観測可能な、累積発見・修正フォールト数の時間発展に着目して、その振舞いから何らかの説得力のある

数理モデルを構築し、得られたデータに適用することにより、当該ソフトウェアの信頼性、つまり「所定の期間内にユーザの期待通りに動作する確率」を評価しようという試みが継続的になされている。これについては現在までの間に、おそらく200以上の確率モデルが提案され、それらの中からいくつかの、比較的取り扱いが簡単なものは実用に供される

ことになったが、しかしながらその精度においては常に疑問符が付される結果となっている。

一般に、これは確率モデルに限ることではないが、モデルを現実の現象に合わせて精緻化しようとするほど、その数学的な取り扱いが困難なものとなる。たとえば、上述したソフトウェア開発の総合テスト工程から得られる累積発見・修正フォールト数の時間変化を解析対象としたモデル、いわゆるソフトウェア信頼度成長モデルに含まれる、NHPP（非同次ポアソン過程）モデルは数学的な取り扱いが簡単であり、実際に（あまり精度の良くないモデルではあるが）ソフトウェアの信頼性評価の現場で利用されるようになってきている。しかしながら、このモデル群は一般のテスト工程では当然生じる、発見されたソフトウェアフォールトの修正の失敗（たとえば、修正作業によりそのテストケースに対しては、当該ソフトウェアは期待通りに動作するようになるが、別のプログラム論理に新たなフォールトを埋め込んでしまうという失敗）をモデルに精密な形で盛り込むことが困難であるという欠点がある。反対に、現実をある程度は精密に表現した数学モデルも数多くモデル化されているが、モデルを実測データに対して適用し、未知パラメータを推定する段階になると、それらは際立った困難さに直面し、実用性にかけるモデルとなることも多い。これらが本研究の背景であり、上述の困難さを打開する方法として、以下に述べるシミュレーションモデルの利用を着想するに至った。

## 2. 研究の目的

上述の状況に鑑み、本研究では、より実用的かつ柔軟な信頼性評価モデルを構築するために、確率モデルの枠組みを一旦離れてみることを考える。すなわち、ソフトウェアのテスト工程から得られるデータの発生メカニズムを確率過程などで捉えるのではなく、もっと自由度の高いアルゴリズムにより表現し、シミュレーションモデルとして現象の表現と再現を行うことを着想した。シミュレーションによる実現現象のモデル化は、数理モデルではそのモデル自身の成立のために必要となる厳しい仮定を緩めることが比較的容易である利点がある。一般に、時系列データを生み出すシミュレーションプログラム（以下ではシミュレータと略す）では、いくつかのシミュレーションのための制御パラメータをもっており、それらの値を適当に設定した上で複数回実行し、シミュレーション結果としての時系列のサンプルを得ることができる。また、その結果を吟味しながら、制御パラメータの値を摂動させるなどして、結果の振る舞いを観察し、いわば感度分析的

に、あるいは制御パラメータの設定→結果の吟味という、一方向的にシミュレーションモデルを利用するのが通例である。

これに対して本研究では、実測されたデータをシミュレータに再現させるために、シミュレータの制御変数をキャリブレーション（較正）する手法を開発することを目的とする。つまり、現実の現象をよく表現するために、確率モデルよりも緩い仮定の下で動作するソフトウェアのデバッグ活動を模擬するシミュレータを作成し、かつ、それが実測されたデータをうまく再現するように、シミュレータの制御変数を逆向きに推定する（本研究ではこのことを較正、あるいはキャリブレーションと呼ぶこととする）手法を開発したいということである。これが実現すれば、従前の確率モデル等をその特別の場合として含み、かつモデル（シミュレータ）に含まれる制御パラメータのキャリブレーションによって実測データに対する将来予測が可能となる新しい信頼性評価手法の枠組みを与えることができる。このことが本研究の目的である。

## 3. 研究の方法

研究の方法としては、シミュレータの開発をコンピュータ上で行うこと、キャリブレーションの方法を検討した上で実装し、試行錯誤を重ねて実測されたソフトウェアのバグデータを再現する方法を模索する手法を採ったが、具体的には以下に述べる手順を採った。

### (1) 基本シミュレータの開発と改良

基本となる（1変数を有する）ソフトウェアデバッグを模擬するシミュレータを拡張することで、ソフトウェアテストの質の問題において常に発生する可能性がありソフトウェアの信頼性評価結果の精度を下げてしまう原因の一つである、不完全デバッグの事象、すなわち発見したソフトウェアプログラム内の論理エラーであるバグを完全には除去できていないあるいは、別のプログラム経路上に作り込んでしまった、という事象を本研究のシミュレータが取り扱うことができるように拡張する。この場合はキャリブレーションの必要がある変数の数が増加するため、それをうまく処理する枠組みが必要となる。

### (2) キャリブレーション法の改良

本研究のシミュレータによる信頼性評価においては、シミュレータに含まれる未知パラメータを実測データによってキャリブレーションするという基本的枠組みがある。このキャリブレーションを遂行する際の評価関数として、以下のようなものを用いている。つまり実測データをシミュレータが精度よく平均的に再現できるときにその値が小さ

く、再現ができなくなるにつれて大きくなるような関数を想定している。この関数は現在の研究の枠組みのままであるならば離散的な、滑らかではない関数となるため、その最小値を与える点、つまりキャリブレーションの最良な結果を与える点を探すとき、通常の微分操作をすることができない。また、その関数の値はシミュレーターを動作させる度に変化するという難点もある（ただしこの特性は信頼性評価結果をある程度の幅を持って推定せしめることができる利点でもあることに注意が必要である）。キャリブレーションの効率化のためにはこの点をうまく克服あるいは緩和する方法が求められる。その具体的な方法を他の分野で用いられる手法の援用について模索することが必要である。本研究はこのような点に対する方針に基づいて行った。

#### 4. 研究成果

以下に、年度ごとに分けて記載することとする。

##### (1) 平成20年度

基本シミュレータの開発と改良に関する成果について述べる。

本年度は従前のシミュレーションモデルの拡張として

- ・不完全デバッグを盛り込んだ場合のシミュレータの改良、
- ・1度のフォールト修正作業において、複数のプログラム論理の変更が生ずるとした場合への拡張、
- ・そもそもの解析対象である、テスト時間の経過に伴って変化する、累積ソフトウェアフォールト発見数のデータが、S字曲線などを描く現象に対応するシミュレータの開発、のそれぞれについて検討し、基礎モデルの構築と評価を行った。これらの成果の一部については、モデルについての多少のbrush upを経たのち、次年度以降において研究成果として発表する予定とした。なお、研究実績に挙げた著書（分担執筆）の内容は、シミュレータによるソフトウェア信頼性評価を行う際の基本的考え方を述べたものであり、関連が深い。この成果を用いることにより、シミュレータのパラメータを実測データに基づいてキャリブレーションした後の信頼性評価手法として、例えばbootstrap法に類似した方法として実装できる可能性があることが指摘された。

##### (2) 平成21年度 キャリブレーション法の改良

具体的な研究の進め方として、この年度では、滑らかではない評価関数がある規範の下で最少とするパラメータを如何にして探索するかについて、ならびに、ソフトウェア信頼性をより狭い範囲に限定したソフトウエ

ア脆弱性に関する評価シミュレータの開発について検討した。前者については効率的な方法を模索するにとどまっているため、今後開発・改良に取り組む予定である。一方、脆弱性評価については、ソフトウェアの脆弱性に関連するセキュリティインシデントデータに基づくことにより、シミュレーションモデルの開発は可能と考えられ、次年度において実装を目指すこととした。なお、研究の遂行において得られた関連する知見については適宜研究発表などを行って公表した。

##### (3) 平成22年度

最終年度である平成22年度は、滑らかではない評価関数がある規範の下で最少とするパラメータを如何にして探索するかについて再度検討した。結果として取り扱っているデータとモデルの構造が比較的小規模・単純であれば特段の工夫をすることなく、実際にソフトウェアの信頼性を表すパラメータの最適値を求めることが可能であることが判明し、この結果を国際会議にて発表した。一方、ソフトウェア脆弱性の評価のモデルについて、同様なアプローチが可能であることは判明しているが、具体的なモデルとデータ解析には至らず、今後の課題としてこれは残された。また、いわゆるソフトウェアのバグがテスト工程において発見されていく際の時系列データの形状が、S字カーブを描くような場合に適合させるため、シミュレーションモデルの記述を工夫する方法論についても研究を行い、一定の成果を得ることができた。これについては国際会議（APARM2010）において公表した。

#### 5. 主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計3件）

① M. Kimura and T. Fujiwara, “Practical Optimal Software Release Decision Making by Bootstrap Moving-Average Quality Control Chart”, International Journal of Software Engineering and Its Applications, Vol. 1, pp. 29-41, 2010. (査読付)

② M. Kimura and T. Fujiwara, “A note on software reliability monitoring by moving-average method based on a bootstrap scheme”, International Journal of Systems Assurance Engineering and Management, Vol. 1, Issue 4, pp. 293-298, 2010. (査読付)

③ M. Kimura and T. Fujiwara, “Practical Optimal Software Release Decision Making by Bootstrap Moving-Average Quality

Control Chart”, International Journal of Software Engineering and Its Applications, Vol. 4, No. 1, pp. 29-41 (2010). (査読付)

[学会発表] (計5件)

① 藤原隆次, 木村光宏, 佐藤吉信, 山田茂: 「ソフトウェアに対する安全度水準計算の一手法」(奨励報文賞受賞), 第40回信頼性・保全性シンポジウム講演論文集, 日本科学技術連盟, 東京 (2010年7月16日).

② M. Kimura, T. Fujiwara, “A Simulation-Based Approach for Software Reliability Assessment”, Proceedings of the 4th Asia-Pacific International Symposium (APARM2010), pp. 341-348, Wellington, New Zealand (2010年12月4日).

③ 藤原隆次, 木村光宏: 「ブートストラップ法を活用したテスト工程におけるソフトウェア信頼性評価手法」, 信学技報, R-2010-35, 大阪 (2010年11月19日).

④ T. Fujiwara, M. Kimura, and S. Yamada, “On the Application and Its Problems of Recent Development Processes for Highly and Safe Software”, Proceedings of the 16th ISSAT International Conference on Reliability and Quality in Design, pp. 45-49, Washington D.C., USA (2010年8月5日).

⑤ M. Kimura and T. Fujiwara, “A new criterion for the optimal software release problems: Moving average quality control chart with bootstrap sampling”, Advances in Software Engineering (Communications in Computer and Information Science), Proceedings of the International Conference on Advanced Software Engineering and Its Applications, pp. 280-287, Jeju Island, Korea (2009年12月11日).

[図書] (計3件)

① 木村光宏: 高信頼化ソフトウェアのための開発手法ガイドブック (分担執筆), IPA/SEC エンタプライズ系ソフトウェア開発力強化推進委員会, pp. 146-149 (2011).

② M. Kimura and T. Fujiwara, “A Bootstrap Software Reliability Assessment Method to Squeeze Out Remaining Faults”, Advances in Computer Science and Information Technology (T.-H. Kim and H. Adeli (Eds.)), Lecture Notes in Computer Science (LNCS6059), pp. 435-446 (2010).

③ M. Kimura, “A linearized growth curve model for software reliability data analysis”, in Recent Advances in Reliability and Quality in Design (H. Pham ed), Chapter 13, pp. 275-290, Springer-Verlag, London, 2008.

## 6. 研究組織

### (1) 研究代表者

木村 光宏 (KIMURA MITSUHIRO)

法政大学・理工学部・教授

研究者番号: 20263486