

# 表計算ソフトウェアによる位相型ソフトウェア信頼性評価ツールの試作

岡村 寛之<sup>†</sup> 土肥 正<sup>†</sup>

<sup>†</sup> 広島大学大学院工学研究科情報工学専攻

〒 739-8527 東広島市鏡山 1-4-1

E-mail: †{okamu,dohi}@rel.hiroshima-u.ac.jp

**あらまし** 本稿では、ソフトウェア信頼度を評価するツール SRATS (Software Reliability Assessment Tool on Spreadsheet) の紹介を行う。SRATS は表計算ソフトウェアのアドインとして機能するツールであり、これまでに数多くの企業での利用実績がある。ここでは、これまで VBA (Visual Basic for Application) を利用して作られていた構成を見直し、C# ならびに WPF (Windows Presentation Foundation) を用いた再構築を行う。また、新たに位相型ソフトウェア信頼性モデルによる評価が行える機能拡張を行う。

**キーワード** ソフトウェア信頼性評価, 表計算ソフトウェア, SRATS, 位相型ソフトウェア信頼性モデル

## Development of a Prototype of Phase-Type Software Reliability Assessment Tool on Spreadsheet

Hiroyuki OKAMURA<sup>†</sup> and Tadashi DOHI<sup>†</sup>

<sup>†</sup> Department of Information Engineering, Graduate School of Engineering, Hiroshima University

1-4-1 Kagamiyama, Higashi-Hiroshima, 739-8527 Japan

E-mail: †{okamu,dohi}@rel.hiroshima-u.ac.jp

**Abstract** This paper introduces SRATS (Software Reliability Assessment Tool on Spreadsheet) that is a tool for software reliability assessment. SRATS is an add-in of a well-known spreadsheet application, and are used by several software development companies. In the paper, we refactor the software architecture for SRATS from VBA (Visual Basic for Application) to C# and WPF (Windows Presentation Foundation). Also, we develop a new feature for the software assessment with phase-type software reliability models.

**Key words** software reliability assessment, spreadsheet, SRATS, phase-type software reliability models

### 1. はじめに

ソフトウェア信頼度成長モデル (SRGM: Software Reliability Growth Model) とは、ソフトウェア開発のテスト工程で観測されるバグ発見のふるまいを確率過程によってモデル化したものであり、バグ発見数データなどからモデルパラメータを推定することにより、将来のバグ発見過程の予測し、ソフトウェアの信頼度の見積もりやリリース判定などを行うことができる。SRGM による研究は 1970 年代から現在まで継続的に行われ、初期のモデルには Jelinski and Moranda [9] によるマルコフ型モデル、Goel and Okumoto [7] による非同次ポアソン (NHPP: Non-Homogeneous Poisson Model) モデルがある。特に、NHPP モデルは数学的な取り扱いが容易であることから、数多くのモデルが構築・提案されている。

一方、個々のモデルの提案とは別に多くの NHPP モデルを包

括的に説明するモデリングフレームワークの議論も行われている。モデリングフレームワークとして最も良く知られているものに Langberg and Singpurwalla [11] によるフレームワーク (以下、LS モデルと呼ぶ) がある。これは、単純な二つのモデル仮定だけで、指数形や S 字形 NHPP モデルなどの代表的なモデルを説明する枠組みであり、実際にこれまでに提案されてきた数多くの NHPP モデルが LS モデルのサブクラスとして記述できる。近年、この枠組みの中で最も広いクラスとして位相型ソフトウェア信頼度成長モデル (PH-SRGM: Phase-Type SRGM) [19, 20, 22] が提案されている。PH-SRGM は数学的な意味で従来のモデルを含むだけでなく、LS モデルにおけるどのモデルも高い精度で近似することが可能である。そのため、従来から指摘されてきたモデル選択などの問題を解決する可能性を有している。

一方で実用面に目を向けると、ソフトウェア開発に携わる開

発者が数学的なソフトウェア信頼度成長モデルを用いてソフトウェア信頼性評価を行うためには、数理的な背景を学習する必要があり学習に関するコストが必要となる。また、データ（ソフトウェアバグ数）が更新される毎にモデルのパラメータ推定および信頼性評価も更新する必要がある。そのため、ソフトウェア信頼性モデルによる信頼性評価では何らかのツールによる支援が重要となる。

実際、ソフトウェア信頼性モデルが提案された初期の頃からツールの開発も行われている。しかしながら、その多くはメンテナンスが不十分であり現在利用できるツールは数多くない。近年、国内において最もよく普及しているツールとして SRATS (Software Reliability Assessment Tool on Spreadsheet) [21, 29] がある。これは、Microsoft (MS) Excel 上のアドインとして開発されたツールであり、最新バージョンの Windows OS や MS Excel での稼働する。しかしながら、ソフトウェアが Windows Form を利用した VBA (Visual Basic for Application) と C++ による DLL (Dynamic Link Library) で構成されており、現在の Windows アプリケーションにおいて主流となっている C# による .NET フレームワークの利用や WPF (Windows Presentation Foundation) と比較すると幾分か古い仕様となっている。

そこで本稿では、SRATS の更新を次の観点で行う。(i) C# でモデル推定部分を再記述することにより将来的な拡張に対応する、(ii) WPF を用いた GUI インタフェースにより、現在の Windows アプリケーションでの標準的な仕様にする。また、今回の更新にあわせて、従来 Java で記述した位相型分布に対する推定プログラム PHSRM [22] を C# に移植することで、SRATS への統合を試みる。

## 2. PH-SRGM

LS モデルは以下の仮定から構築される。

- ソフトウェア内のバグ数は有限であり、平均  $\omega$  のポアソン分布に従う。
- 各バグの発見時刻は互いに独立であり、同一の確率分布に従う。

いま、バグの発見時刻分布の累積分布関数を  $F(t)$  とすると、時刻  $t$  までの累積発見バグ数  $N(t)$  は

$$P(N(t) = x) = \frac{(\omega F(t))^x}{x!} e^{-\omega F(t)} \quad (1)$$

となる。上記の確率関数は平均値関数  $E[N(t)] = \omega F(t)$  の NHPP となっている。つまり、LS モデルでは個々の NHPP モデルにおける違いをバグ発見時刻分布の違いとして表現している。実際に  $F(t)$  に対して具体的な確率分布の累積分布関数を代入することにより、既存のモデルの多くを表現できる。表 1 は SRATS に実装されている代表的なバグ発見時刻分布と NHPP モデルの対応を表している。

PH-SRGM はバグ発見時間分布に位相型分布 (PH: Phase-Type Distribution) [16] を適用したモデルとなっている。位相型分布は吸収状態をもつ連続時間マルコフ連鎖において吸収状態へ到達するまでの時間に対する確率分布である。指数分布の

表 1: モデルとバグ発見時間分布の対応。

Model	Distribution
Goel and Okumoto model [7]	Exponential
Delayed S-shaped model [27]	Gamma
$k$ -stage Erlangian model [10, 28]	
Littlewood NHPP model [13]	Pareto
Truncated-normal model [23]	Truncated normal
Log-Normal model [1, 23]	Log-normal
Infection S-shaped model [17]	Truncated logistic
Log-Logistic model [8]	Log-logistic
Gompertz model [18, 30]	Truncated extreme-value max
Gompertz model (min) [18]	Truncated extreme-value min
Frechet model [18]	Log-extreme-value max
Generalized exponential model [6]	Weibull (Log-extreme-value min)

表 2: モデルと位相型分布の対応。

Model	PH Representation
Goel and Okumoto model [7]	Exponential
Delayed S-shaped model [27]	2-Erlang
$k$ -stage Erlangian model [10, 28]	$k$ -Erlang
Hyperexponential model [12, 15]	Hyperexponential
Testing-domain dependent model [5]	Hypoexponential

拡張となっているため、指数分布、アーラン分布、超指数分布、劣指数分布をサブクラスとして包括する。また、非負の確率変数に対する分布に対して稠密となっている [3]。つまり、任意の確率分布を任意の精度で近似することが可能である。具体的に、PH-SRGM の確率関数は以下のように与えられる。

$$P(N(t) = n) = \frac{(\omega F_{PH}(t))^n}{n!} e^{-\omega F_{PH}(t)}, \quad (2)$$

$$F_{PH}(t) = 1 - \alpha \exp(\mathbf{T}t)\mathbf{1}. \quad (3)$$

ここで、 $\alpha$ ,  $\mathbf{T}$  は吸収状態を持つ連続時間マルコフ連鎖の初期確率ベクトルならびに無限小生成行列である。また、 $\mathbf{1}$  はすべての要素が 1 の列ベクトルである。連続時間マルコフ連鎖の状態数 (位相数と呼ぶ) と状態遷移率 (位相構造) によって異なる表現のモデルとなる。位相型分布の性質から、PH-SRGM は表 2 に示す既存の NHPP モデルをサブクラスとして包括する。また、LS モデルのサブクラスを任意の精度で近似することができる。

PH-SRGM では位相数と位相構造によってモデルの表現力が決まる。一般的な PH-SRGM (位相構造に制約がない PH-SRGM) が最も表現力が高いが冗長な表現を含むため位相構造を限定したものが利用するが後述する統計処理などにおいて効率が良い。そのため、文献 [19, 20] では、Canonical PH (以下 CPH) と超アーラン分布 (以下 HEr) の構造を持つ PH-SRGM を提案している。CPH は非周期的な構造をもつ連続時間マルコフ連鎖に対する標準形となっている [4]。つまり、任意の非周期的な構造をもつ位相型分布と等価な表現力を持つ。CPH-SRGM の確率関数は以下のようになる [19]。

$$P(N(t) = n) = \frac{(\omega F_{CPH}(t))^n}{n!} e^{-\omega F_{CPH}(t)}, \quad (4)$$

$$F_{CPH}(t) = 1 - \alpha \exp(\mathbf{T}t)\mathbf{1}, \quad (5)$$

$$\alpha = (\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_n), \quad (6)$$

$$\mathbf{T} = \begin{pmatrix} -\lambda_1 & \lambda_1 & & & \\ & \ddots & \ddots & & \\ & & -\lambda_{n-1} & \lambda_{n-1} & \\ & & & -\lambda_n & \end{pmatrix}. \quad (7)$$

ただし  $\sum_{i=1}^n \alpha_i = 1$  ならびに  $\lambda_1 \leq \cdots \leq \lambda_n$  である。

HEr は CPH よりも表現力は落ちるものの、行列指数関数を直接計算する必要がないため計算が容易になる利点がある。具体的な HEr-SRGM の確率関数は以下のようにになる [20]。

$$P(N(t) = n) = \frac{(\omega F_{HEr}(t))^n}{n!} e^{-\omega F_{HEr}(t)}, \quad (8)$$

$$F_{HEr}(t) = \sum_{i=1}^m p_i \int_0^t \frac{\beta_i^{r_i} x^{r_i-1} e^{-\beta_i x}}{(r_i - 1)!} dx, \quad (9)$$

$$\sum_{i=1}^m p_i = 1.$$

ここで式 (9) は次数  $r_1, \dots, r_m$  のアーラン分布の累積分布関数の  $p_1, \dots, p_m$  による加重和である。HEr-SRGM の位相数は  $\sum_{i=1}^m r_i$  で与えられることに注意する。

### 3. 統計的推定

#### 3.1 パラメータ推定

いま、バグ検出数データ  $\mathcal{I} = \{(t_1, n_1), \dots, (t_k, n_k)\}$  を考える。ここで、 $t_1, \dots, t_k$  は累積のテスト時間、 $n_i, i = 1, \dots, k$  は時間間隔  $[t_{i-1}, t_i]$  で発見されたバグ数を表す。このとき、LS モデルの対数尤度は以下のように与えられる。

$$\begin{aligned} \log \mathcal{L}(\theta|\mathcal{I}) &= \sum_{i=1}^k n_i \log \omega + \sum_{i=1}^k n_i \log(F(t_i) - F(t_{i-1})) \\ &\quad - \sum_{i=1}^k \log n_i! - \omega F(t_k). \end{aligned} \quad (10)$$

つまり、パラメータ推定問題は上記の対数尤度を最大にするモデルパラメータを見つける最適化問題に帰着される。しかしながら、位相数の大きい PH-SRGM ではモデルパラメータが多くなるため、上記を最大にするパラメータの導出が難しくなる。この問題に対して Okamura and Dohi [19, 20] は EM アルゴリズムによる効率的かつ高速なパラメータ推定手法の提案を行っている。

EM アルゴリズムとは未観測データを含むモデルに対する最尤推定値を算出するための汎用的な手法であり、大域的な収束性を持つためパラメータが多いモデルに対しても安定したパラメータ推定アルゴリズムを構築できる。具体的に、文献 [22] では (i) 各バグ発見時刻までの位相変化、(ii) 観測時間時点での残存バグ数、(iii) 各時間区間での詳細なバグ発見時刻が未観測とすることで、位相数  $n$  が与えられたもとの EM アルゴリズムを構築している。

**Step 0:** 初期パラメータ  $\omega, \alpha, \mathbf{T}$  を決定する。

**Step 1:** 以下の微分方程式を解き、各時刻  $t_i, i = 1, \dots, k$  における  $\mathbf{f}(t), \tilde{\mathbf{f}}(t), \tilde{\mathbf{b}}(t), \tilde{\gamma}_i(t)$  を算出する。

$$\mathbf{f}(t) = \alpha \exp(\mathbf{T}t), \quad (11)$$

$$\tilde{\mathbf{f}}(t) = \alpha(-\mathbf{T})^{-1} \exp(\mathbf{T}t), \quad (12)$$

$$\tilde{\mathbf{b}}(t) = \exp(\mathbf{T}t)\mathbf{1}, \quad (13)$$

$$\tilde{\gamma}_i(t) = \int_0^t [\mathbf{f}(u)]_i \tilde{\mathbf{b}}(t-u) du + [\tilde{\mathbf{f}}(t)]_i \mathbf{1}. \quad (14)$$

ここで、 $[\cdot]_i$  はベクトルの  $i$  番目の要素を表す。

**Step 2:** 次の更新式に従ってパラメータを更新する。

$$\omega \leftarrow \sum_{u=1}^k n_u + \omega \mathbf{f}(t_k)\mathbf{1}, \quad (15)$$

$$\alpha_i \leftarrow \frac{\alpha_i \left\{ \sum_{u=1}^k \frac{n_u [\tilde{\mathbf{b}}(t_{u-1}) - \tilde{\mathbf{b}}(t_u)]_i}{\pi(\tilde{\mathbf{b}}(t_{u-1}) - \tilde{\mathbf{b}}(t_u))} + \omega [\tilde{\mathbf{b}}(t_k)]_i \right\}}{\sum_{u=1}^k n_u + \omega \mathbf{f}(t_k)\mathbf{1}}, \quad (16)$$

$$\mu_{i,j} \leftarrow \frac{\mu_{i,j} \left\{ \sum_{u=1}^k \frac{n_u [\tilde{\gamma}_i(t_{u-1}) - \tilde{\gamma}_i(t_u)]_j}{\alpha(\tilde{\mathbf{b}}(t_{u-1}) - \tilde{\mathbf{b}}(t_u))} + \omega [\tilde{\gamma}_i(t_k)]_j \right\}}{\sum_{u=1}^k \frac{n_u [\tilde{\gamma}_i(t_{u-1}) - \tilde{\gamma}_i(t_u)]_i}{\alpha(\tilde{\mathbf{b}}(t_{u-1}) - \tilde{\mathbf{b}}(t_u))} + \omega [\tilde{\gamma}_i(t_k)]_i}, \quad (17)$$

$$\tau_i \leftarrow \frac{\tau_i \left\{ \sum_{u=1}^k \frac{n_u [\tilde{\mathbf{f}}(t_{u-1}) - \tilde{\mathbf{f}}(t_u)]_i}{\alpha(\tilde{\mathbf{b}}(t_{u-1}) - \tilde{\mathbf{b}}(t_u))} + \omega [\tilde{\mathbf{f}}(t_k)]_i \right\}}{\sum_{u=1}^k \frac{n_u [\tilde{\gamma}_i(t_{u-1}) - \tilde{\gamma}_i(t_u)]_i}{\alpha(\tilde{\mathbf{b}}(t_{u-1}) - \tilde{\mathbf{b}}(t_u))} + \omega [\tilde{\gamma}_i(t_k)]_i}. \quad (18)$$

ここで  $\mu_{i,j}$  は  $\mathbf{T}$  の  $i, j$  要素、 $\tau_i$  は  $\boldsymbol{\tau} = -\mathbf{T}\mathbf{1}$  の  $i$  番目の要素を表す。

**Step 3:** 停止条件を満たしていれば終了。そのときの推定値は  $\omega, \alpha, \mathbf{T}, \boldsymbol{\tau}$  で与えられる。停止条件を満たしていないならば Step 1 へ。

特に、文献 [22] では式 (11) から (14) における計算を一様化手法を応用して高速化する手法についても言及している。

同様に HEr-SRGM に対する推定は、(i) バグ発見時刻が従うアーラン分布、(ii) 観測時間時点での残存バグ数、(iii) 各時間区間での詳細なバグ発見時刻が未観測データとすることで各アーラン分布の次数が与えられたもとの EM アルゴリズムを構築している。

**Step 0:** 初期パラメータ  $\omega, p_1, \dots, p_m, \beta_1, \dots, \beta_m$  を決定する。

**Step 1:** 以下の更新式に従ってパラメータを更新する。

$$\omega \leftarrow \sum_{u=1}^k n_u + \omega \sum_{i=1}^m p_i \bar{F}_i(t_k), \quad (19)$$

$$p_i \leftarrow \frac{\sum_{u=1}^k \frac{n_u p_i (F_i(t_u) - F_i(t_{u-1}))}{\sum_{i=1}^m p_i (F_i(t_u) - F_i(t_{u-1}))} + \omega p_i \bar{F}_i(t_k)}{\sum_{u=1}^k n_u + \omega \sum_{i=1}^m p_i \bar{F}_i(t_k)}, \quad (20)$$

$$\beta_i \leftarrow \frac{r_i \left\{ \sum_{u=1}^k \frac{n_u p_i (F_i(t_u) - F_i(t_{u-1}))}{\sum_{i=1}^m p_i (F_i(t_u) - F_i(t_{u-1}))} + \omega p_i \bar{F}_i(t_k) \right\}}{\sum_{u=1}^k \frac{n_u r_i s_u p_i (F'_i(t_u) - F'_i(t_{u-1}))}{\sum_{i=1}^m p_i (F_i(t_u) - F_i(t_{u-1}))} + \omega r_i p_i \bar{F}'_i(t_k)} \quad (21)$$

ここで  $s_u = t_u - t_{u-1}$  である。また、 $F_i(t)$  と  $F'_i(t)$  はそれぞれパラメータ  $(r_i, \beta_i)$  ならびに  $(r_i + 1, \beta_i)$  のアーラン分布の累積分布関数を表す。さらに  $\bar{F}(t) = 1 - F(t)$  とする。

**Step 2:** 停止条件を満たしていれば終了。そのときの推定値は  $\omega, p_1, \dots, p_m, \beta_1, \dots, \beta_m$  で与えられる。停止条件を満たしていないならば Step 1 へ。

さらに文献 [20, 22] では  $r_i$  の推定に関して、与えられた位相数  $n$  に対して  $\sum_{i=1}^m r_i = n$  となるような  $r_i$  の組み合わせをすべて列挙し、対数尤度を最大にする  $r_i$  の組み合わせを見つける手法を取り入れている。

### 3.2 位相数の決定

PH-SRGM では位相数が増加するにつれてパラメータ数が増加するため最大対数尤度が単調に増加する。換言すれば、位相数の大きい PH-SRGM はデータに対する過適合を引き起こす。そのため、パラメータ数（モデル自由度）を考慮した何らかの指標にもとづいて位相数を決定する必要がある。一般的によく利用される指標として情報量規準（IC: information criterion）がある。IC は真の分布と推定された分布とのカルバックライブラー情報量の近似から得られる指標であり、真の分布からのある種の（汎）距離を表している。つまり、IC が小さい値をとるほど良いモデル（真の分布に近いモデル）となり、最尤推定における過適合を防ぐことができる。

一般的に、最尤推定法の枠組みにおける IC は最大対数尤度とバイアスで構成される。しかしながら、一般に真の分布はわからないためバイアス  $b$  の値を何らかの手法で近似的に推定する必要がある。様々な情報量規準は、バイアス  $b$  の推定手法の違いで区別することができる。一般的によく知られている AIC [2] は以下の式で算出される。

$$\text{AIC} = -2 \max_{\theta} \log \mathcal{L}(\theta) + 2\text{DF}. \quad (22)$$

ここで、 $\max_{\theta} \log \mathcal{L}(\theta)$  は最大対数尤度である。これはバイアスをモデルの自由度（DF）で与えている。また、BIC [25] はベイズの事後確率に基づいた情報量規準であり、次の式で算出される。

$$\text{BIC} = -2 \max_{\theta} \log \mathcal{L}(\theta) + \text{DF} \log k. \quad (23)$$

ここで  $k$  はデータ数を表す。AIC, BIC いずれもモデル自由度が必要となるが、文献 [22] では位相数  $n$  の CPH-SRGM に対するモデル自由度を自由パラメータ数  $2n$  で与えている。一方 HEr-SRGM に対しては自由度を  $2m$  としている。これは、位相数  $n$  ならびにアーラン分布の次数  $r_1, \dots, r_m$  が与えられたもとの、自由パラメータ数に対応している。実際には位相数  $n$  が与えられたとしてもアーラン分布の次数  $r_1, \dots, r_m$  を選ぶ

ため、 $2m$  によるモデル自由度は少なく見積もっていることに注意する。

また、情報量規準以外の適合性として MSE (Mean Squared Error) [24] もよく利用される。これは非線形回帰モデルによく用いられる指標であり、以下のように定義する。

$$\text{MSE} = \frac{1}{k - \text{DF}} \sum_{u=1}^k (\omega F(t_u) - \sum_{l=1}^u n_l)^2. \quad (24)$$

## 4. 信頼性評価ツール

### 4.1 ツール概要

本稿では、PH-SRGM によるソフトウェア信頼性評価が行えるツールの試作を行う。特に、表計算ソフトウェアのアドインとして動作する SRATS に対して PH-SRGM を利用できる拡張を行うと同時に現在の動作環境に合わせたソフトウェアのメンテナンスも行う。作成するツールの機能概要は以下の通りである。

- MS Excel 2010 以降のアドインとして開発
- MS Excel 上のデータを利用
- 既存の SRATS で利用できる 11 種類のモデル
- CPH-SRGM, HEr-SRGM の追加
- ツール上での推定結果のプレビューを追加

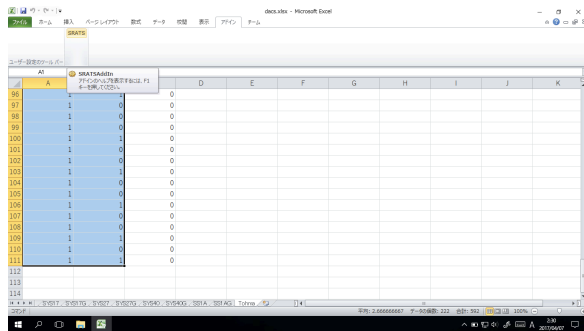
MS Excel 2010 以降での動作を前提とするため、従来の VBA (Visual Basic for Application) と C++ による実装から、C# ならびに VSTO (Visual Studio for Tools) によるアドインとして開発する。特に、WPF (Windows Presentation Foundation) ならびに MVVM (Model-View-ViewModel) による再設計を行うことで現在の Windows アプリケーションに沿った開発手法を取り入れている。これにより、C# で記述されたモデル推定部分の再利用が容易になっている。さらに、Sparrow Chart WPF を利用した推定結果のグラフによるプレビュー機能を追加している。作成したツールは MIT ライセンスで GitHub にて公開している<sup>(注1)</sup>。インストールなどについては、Visual Studio が提供する ClickOnce によるインストーラを提供している。

### 4.2 使用例

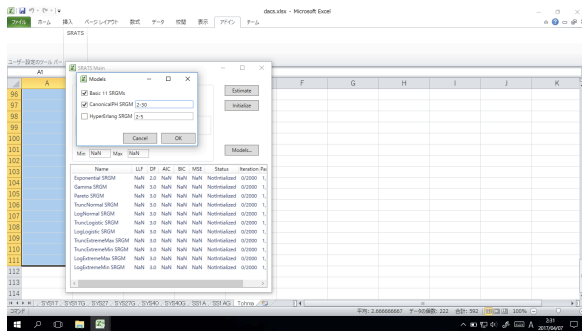
ここでは、実際の開発現場で計測されたバグ発見データを用いた分析例を示す。データは Tohma et al. [26] による約 200 モジュールからなるリアルタイム制御に対するテストで得られたバグ発見個数データであり、111 の観測で 481 個のバグが発見されている。データにはいわゆるチェンジポイントがいくつかあり、単純な SRGM では推定が難しいデータである。

SRATS ではバグ個数データについては時間間隔と発見個数の 2 列からなるデータを利用する。MS Excel を起動後に対応する 2 列のデータを選択し、アドインメニューから SRATS を起動する（図 1(a)）。SRATS では予め基本となる 11 種類の SRGM が登録されているが、PH-SRGM の利用には Models ボタンによるモデルの追加が必要となる（図 1(b)）。モデルの追加では CPH-SRGM と HEr-SRGM が選択可能であり、位相数

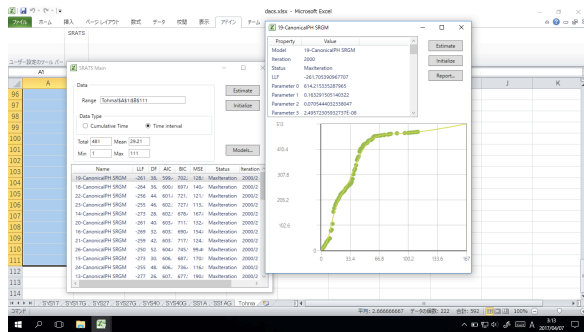
(注1) : <https://github.com/rellab/SRATS/>



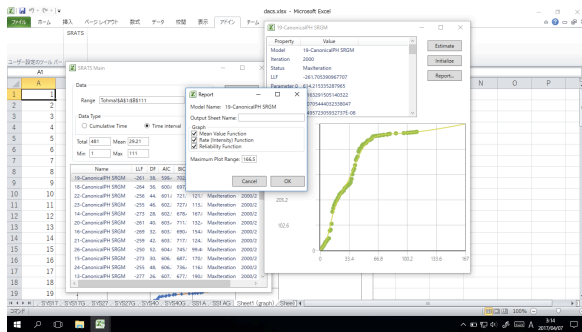
(a) データの選択と SRATS の起動.



(b) PH-SRGM の追加.



(c) 推定結果の表示.



(d) 評価結果のレポート出力.

図 1: SRATS の動作例.

の指定欄では推定する位相数をテキストにより入力する. 推定ならびにレポートの出力については, 旧バージョンの SRATS と同じで, Estimate による全モデルのパラメータ推定ができる. また開発したバージョンでは LLF, AIC, BIC, MSE の各列ラベルをクリックすることで昇順・降順のソートを行うことができる. さらに, リストのダブルクリックによるパラメータならびに推定された平均値関数の概略が表示される (図 1(c)). また, 従来の SRATS 同様, Report ボタンをクリックすることにより, Excel シートに結果の出力を行う (図 1(d)).

## 5. 関連研究

ソフトウェア信頼度成長モデルを用いたツールはこれまでに数多く開発されてきたが, 現在利用できるツールは数少ない. 1990 年前後には SRET [31], SOREM [32] が発表されているがいずれも入手不可能となっている. また, 企業が提供するツールとして, 東芝による SQATP, 富士通周辺機器による SafeMan, 構造計画研究所による信頼度成長曲線評価ツールがあったがいずれも現在では入手できなくなっている. 海外では, Duke 大学により SREPT が開発されているが利用環境の構築が現在では難しくなっている. 一方で, 利用できるツールとしては, Lyu によって編纂編集されたソフトウェア信頼性のハンドブック [14] の CD-ROM を通じて CASRE, SMERFS, AT&T Toolkit, SoftRel が入手できる. また, 田村らによって開発されたツールが <http://www.comm.tcu.ac.jp/~tamura/> にまとめられている.

## 6. まとめと今後の課題

本稿では, 表計算ソフトウェアのアドインとして利用できる SRATS の紹介を行った. 特に, PH-SRGM を扱うための機能拡張を行った. また, 従来の VBA による構成から C# による WPF を用いたアプリケーションとして再構築を行った. 作成したツールは GitHub より入手でき VSTO によるアドインとして手軽にインストールならびに利用が行える. さらに, モデルの推定や評価値の算出が C# によるライブラリ化されており, 多くの .NET アプリケーションに組み込むことができる. 旧版の SRATS は既に国内の多くの企業に利用されており, 現在の Excel の仕様に沿った今回のアップデートは継続的な利用を保証する上で非常に重要である. 今後は, ソフトウェアメトリクスを利用するソフトウェア信頼性モデルを扱えるアプリケーションの開発ならびに, JavaScript などの技術を用いた Excel Web アドインへの対応などが考えられる.

## 文 献

- [1] J. A. Achcar, D. K. Dey, and M. Niverthi. A Bayesian approach using nonhomogeneous Poisson processes for software reliability models. In A. P. Basu, K. S. Basu, and S. Mukhopadhyay, editors, *Frontiers in Reliability*, pages 1–18. World Scientific, Singapore, 1998.
- [2] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, *Proc. 2nd Int. Symp. Inform. Theory*, pages 267–281, 1973.
- [3] S. Asmussen and G. Koole. Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability*, 30:365–372, 1993.
- [4] A. Cumani. On the canonical representation of homo-

- geneous Markov processes modelling failure-time distributions. *Microelectronics and Reliability*, 22:583–602, 1982.
- [5] T. Fujiwara and S. Yamada. Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit. In *Proceedings of the 7th ISSAT International Conference on Reliability and Quality in Design*, pages 36–40, 2001.
  - [6] A. L. Goel. Software reliability models: Assumptions, limitations and applicability. *IEEE Transactions on Software Engineering*, SE-11:1411–1423, 1985.
  - [7] A. L. Goel and K. Okumoto. Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability*, R-28:206–211, 1979.
  - [8] S. S. Gokhale and K. S. Trivedi. Log-logistic software reliability growth model. In *Proc. 3rd IEEE Int'l. High-Assurance Systems Eng. Symp. (HASE-1998)*, pages 34–41. IEEE CS Press, 1998.
  - [9] Z. Jelinski and P. B. Moranda. Software reliability research. In W. Freiberger, editor, *Statistical Computer Performance Evaluation*, pages 465–484. Academic Press, New York, 1972.
  - [10] T. M. Khoshgoftaar. Nonhomogeneous Poisson processes for software reliability growth. In *Proceedings of the International Conference on Computational Statistics (COMPSTAT)*, pages 13–14, 1988.
  - [11] N. Langberg and N. D. Singpurwalla. Unification of some software reliability models. *SIAM Journal on Scientific Computing*, 6(3):781–790, 1985.
  - [12] J. C. Laprie, K. Kanoun, C. Béounes, and M. Kaâniche. The KAT (Knowledge-Action-Transformation) approach to the modeling and evaluation of reliability and availability growth. *IEEE Transactions on Software Engineering*, 17(4):370–382, 1991.
  - [13] B. Littlewood. Rationale for a modified duane model. *IEEE Transactions on Reliability*, R-33(2):157–159, 1984.
  - [14] M. R. Lyu, editor. *Handbook of Software Reliability Engineering*. McGraw-Hill, New York, 1996.
  - [15] K. Matsumoto, K. Inoue, T. Kikuno, and K. Torii. Experimental evaluation of software reliability growth models. In *Proc. 18th Int'l Symp. on Fault-Tolerant Computing*, pages 148–153, 1988.
  - [16] M. F. Neuts. Probability distribution of phase type. In *Liber Amicorum Professor Emeritus H. Florin*, pages 173–206. University of Louvain, 1975.
  - [17] M. Ohba. Inflection S-shaped software reliability growth model. In S. Osaki and Y. Hatoyama, editors, *Stochastic Models in Reliability Theory*, pages 144–165. Springer-Verlag, Berlin, 1984.
  - [18] K. Ohishi, H. Okamura, and T. Dohi. Gompertz software reliability model: estimation algorithm and empirical validation. *Journal of Systems and Software*, 82(3):535–543, 3 2009.
  - [19] H. Okamura and T. Dohi. Building phase-type software reliability models. In *Proceedings of 17th International Symposium on Software Reliability Engineering (ISSRE'06)*, pages 289–298, Los Alamitos, 11 2006. IEEE Computer Society Press.
  - [20] H. Okamura and T. Dohi. Hyper-Erlang software reliability model. In *Proceedings of 14th Pacific Rim International Symposium on Dependable Computing (PRDC'08)*, pages 232–239, 2008.
  - [21] H. Okamura and T. Dohi. SRATS: Software reliability assessment tool on spreadsheet (experience report). In *Proceedings of the 24th International Symposium on Software Reliability Engineering (ISSRE 2013)*, pages 100–117. IEEE CPS, 2013.
  - [22] H. Okamura and T. Dohi. Phase-type software reliability model: Parameter estimation algorithms with grouped data. *Annals of Operations Research*, 244:177–208, 2016.
  - [23] H. Okamura, T. Dohi, and S. Osaki. Software reliability growth models with normal failure time distributions. *Reliability Engineering and System Safety*, 116:135–141, 2013.
  - [24] H. Pham. *Software Reliability*. Springer, Singapore, 2000.
  - [25] G. Schwarz. Estimating the dimension of a model. *Ann. Stat.*, 6:461–464, 1978.
  - [26] Y. Tohma, Y. Yamano, H. Ohba, and R. Jacoby. The estimation of parameters of the hypergeometric distribution and its application to the software reliability growth model. *IEEE Transactions on Software Engineering*, 17:483–489, 1991.
  - [27] S. Yamada, M. Ohba, and S. Osaki. S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability*, R-32:475–478, 1983.
  - [28] M. Zhao and M. Xie. On maximum likelihood estimation for a general non-homogeneous Poisson process. *Scandinavian Journal of Statistics*, 23:597–607, 1996.
  - [29] 岡村寛之, 安藤光昭, 土肥正, 表計算ソフトウェアによるソフトウェア信頼性評価ツール (SRATS) の開発, 電子情報通信学会論文誌 D-I, J88-D-I(2):205–214, 2 2005.
  - [30] 山田茂, ゴンベルツ曲線を用いた確率的ソフトウェア信頼度成長モデル, 情報処理学会論文誌, 33(7):964–969, 1992.
  - [31] 山田茂, 磯崎龍史, 尾崎俊治, ソフトウェア信頼性評価ツール (SRET) の作成, 電子情報通信学会論文誌, J72-D(1):24–32, 1989.
  - [32] 上村松男, 阿部勝徳, 山田茂, ソフトウェア品質を予測する信頼度評価技法 SOREM, 日本オペレーションズ・リサーチ学会春季研究発表会アブストラクト集, pages 156–157, 1991.