

ソフトウェア信頼性モデルを応用した大学等における 原稿作成プロセスの定量的評価

土井 崇[†]

奥田 隆史[†]

井手口 哲夫[†]

田 学軍[†]

愛知県立大学 情報科学部 情報科学科[†]

1 はじめに

大多数の理工系学生は研究室に属し、教員等と共著の学会原稿を作成する。その際、学生は提出期限内に、それまでの講義レポートなどで要求されてきた水準以上の質（他人に内容を一義に理解してもらう）を有する原稿を作成することになる。そのため、大多数の研究室では、学生・教員間での共同での**原稿作成プロセス**（まずは学生が初稿を作成し、教員へ提出する。教員は原稿の修正箇所をマーク・修正し返却し、学生は原稿を修正・再提出する。これらを原稿完成まで繰り返す）が生まれる。

このプロセスの問題点は、プロセス終了時間が、当該学生の実績作成・修正スキルや意欲等の**個人差**に強く左右されるため、計画的に進まないことである。そこで、本研究では学生を持つ複数の原稿修正スキルを想定し、教員の推敲・修正間隔と学生の実績修正スキルが完成時間に与える影響を、ソフトウェア信頼性モデル [1] を応用した確率モデルを用いて検討する。

本稿では、まず、2節で、学生と教員のそれぞれ1名でおこなう原稿作成プロセスについて述べる。3節で本研究で想定する原稿作成プロセスのモデルを定義する。4節でエラー率とソフトウェア信頼性モデルを応用した学生の実績修正スキルの確率モデルを提案する。次に、5節において、学生の実績修正スキルが原稿の作成にどのような影響を与えるかを、シミュレーションにより検証する。6節では、シミュレーション結果の改善方法についての検討・考察をおこなう。最後に7節でまとめる。

2 原稿作成プロセス

本研究で想定する原稿は、他人に内容を一義に理解してもらうというという明確な意図を持った文書に属するものであり、文章構成は定型かつ明確に構成され、一般には理工系文書 [2] と呼ばれるものである。そのため、想定原稿の構成は、5つのセクション（表題 H、著者抄録 Ab、序論 In、本論 Mi、結論 Co）に分割できるとする。

本研究の原稿作成プロセスは、理工系の研究室に属する学生と学生の担当教員で行う1対1の双方向型指導である。この指導では、まず学生が原稿を作成し、教員が学生作成原稿の確認をおこない**修正・不足箇所等**

の指摘をする。学生は指摘箇所はむろん関連箇所を修正し、再度教員へ修正原稿を提出する。このプロセスを確率モデルとして解釈すると、提出される学生作成原稿は学生の実績作成・修正スキルに依存した**原稿エラー率**を持ち、原稿エラー率が一定水準になるまでこの原稿作成プロセスを繰り返すことと同義となる。なお、教員の**修正・不足箇所指摘スタイル**には、学生に自己推敲のポイントだけを示す場合と、点検・添削の結果を学生に示す場合とがあるとする。

3 原稿作成プロセスの評価モデル

本研究の原稿作成プロセスモデルを図1に示す。

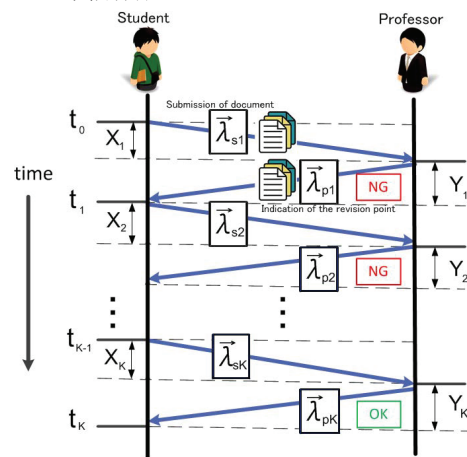


図1 原稿作成プロセスモデル

本モデルでは学生が原稿作成を開始した時刻を t_0 、原稿エラー率が一定水準になる K 回目の提出の時刻を t_K とする。学生が教員に原稿を i ($1 \leq i \leq K$) 回目の提出し学生に原稿が返却された時刻を t_i とする。なお、時刻 t_i での原稿は、各セクションのエラー率を要素とする**学生原稿エラー率ベクトル** $\vec{\lambda}_{si} = (\lambda_{H_i}, \lambda_{Ab_i}, \lambda_{In_i}, \lambda_{Mi_i}, \lambda_{Co_i})$ を持つ。一方、教員の返却原稿には、 $\vec{\lambda}_{si}$ と同様の構成で、 $i+1$ 回目学生提出する原稿の目標エラー率を示す**目標エラー率ベクトル** $\vec{\lambda}_{pi}$ が伴うことになる。本研究の原稿作成手順は次の1~3とする。

1. 学生が作成したエラー率ベクトル $\vec{\lambda}_{si}$ の原稿を、教員に i 回目の提出をする。学生の i 回目の学生の実績作成時間 X_i は、平均作成時間 μ_s^{-1} の指数分布に従う。
2. 原稿は教員が確認をし、学生に対して指導をおこない次回の目標エラー率ベクトル $\vec{\lambda}_{pi}$ を学生に掲示して返却する。教員の i 回目原稿確認時間 Y_i は、平均原稿確認時間 μ_p^{-1} の指数分布に従う。
3. 学生は、学生の修正スキルに応じた修正を原稿にお

Quantitative evaluation of academic-document production process using software reliability models

[†]Takashi DOI, Takashi OKUDA, Xuejun TIAN, Tetsuo IDEGUCHI

[†]Department of Information Science and Technology, Faculty of Information Science and Technology, Aichi Prefectural University

こない, エラー率ベクトル $\vec{\lambda}_{si}$ の要素のすべてが一定水準になる $\vec{\lambda}_{sK}$ まで, 手順の 1~3 を繰り返す.

4 原稿のエラー率と学生の修正スキル

本研究では想定原稿を, 複数文章 (コード) から構成されるソフトウェアのプログラムコードのように捉え, 学生作成原稿の修正数 (エラー・バグ数) は, プログラムコードにおけるバグ件数に似た性質があるものとする. つまり, 原稿のエラー数と修正数の関係を, ソフトウェアに内在しているバグ発見数モデルを利用して表現する. ここで, 原稿の各セクションの初期エラー数 a と, 時間区間 $(0, t]$ までに発見された各セクションのエラー数を確率変数 $N(t)$ とし, 原稿の各セクションのエラー率 λ は,

$$\lambda = 1 - \frac{N(t)}{a}$$

となる. なお, ソフトウェア信頼性評価モデルの NHPP (nonhomogeneous Poisson process) モデル [3] を用いると $N(t)$ は,

$$Pr \{N(t) = n\} = \frac{\{H(t)\}^n}{n!} e^{-H(t)} \quad (n = 0, 1, 2, \dots)$$

$$H(t) = \int_0^t h(x) dx \quad (t \geq 0),$$

に従う. ただし, $H(t)$ は, ソフトウェア信頼性評価モデルでは $N(t)$ の時間区間 $(0, t]$ において発見される総期待エラー数に相当するものであり, 本稿では $H(t)$ を学生の修正スキルとし, 以下の (1)~(5) を考える. なお, 修正スキルは各セクションで独立しているものとする.

$$H(t_i) = a - b(a - N(t_{i-1})) \quad (1)$$

$$H(t) = \frac{a}{c} t \quad (2)$$

$$H(t) = a(1 - e^{-dt}) \quad (3)$$

$$H(t) = \frac{a}{1 + e^{-p(t-r)}} \quad (4)$$

$$H(t) = a \cdot q(e^{-p(t-r)}) \quad (5)$$

なお, $N(0) = H(0) = 0$, a を初期エラー数, b, c, d, p, q, r は学生固有のパラメータである. 学生の修正スキルは, 学生の原稿に対する修正数を表している. 式 (1) は学生の修正数が前回のエラー数にのみ依存する場合, 式 (2) は学生の修正数が一定数の場合, 式 (3) は学生の修正数が残存エラー 1 個当たりのエラー発見率に依存して変化する場合を表現している. 式 (4) と式 (5) は, ロジスティック曲線とゴンペルツ曲線の S 字形成長曲線を表す.

5 数値例

各修正スキル ((1)~(5)) に対する平均エラー率との関係を図 2 に示す. 平均エラー率は 5 回の離散シミュレーション結果の平均である (乱数発生は離散事象シミュレーションパッケージ Csim20[4] を利用). なお, シミュレーションの条件は, 著者の原

稿作成期間が平均で 75 日であることから, 75 日で原稿エラーが基準値以下になるように各変数の値は $\mu_s^{-1}=4[\text{day}]$, $\mu_p^{-1}=1[\text{day}]$, $a=100$, $b=0.75$, $c=60$, $d=0.08$, $p=0.15$, $q=0.5$, $r=30$ と定めた.

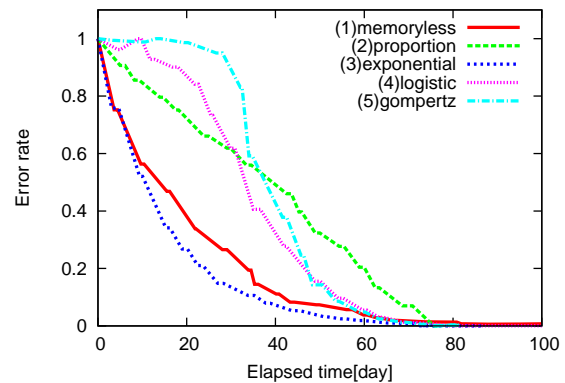


図 2 各修正スキルのエラー率の推移

6 検討・考察: 原稿作成指導への活用

大学における研究指導は学生の自主性を尊重した上で, 高い学習効果を得る必要がある. そのためには, 学生のスキルにあわせて具体的な計画を立てる必要がある [5]. 図 2 で得られた結果を基に指導へのフィードバック例を検討する.

例えば, 式 (1) と式 (3) のような修正をする学生に対しては, エラー率が減少するのは速いが時間とともに修正量が減るため, 共同学習やグループ学習の機会を提供することで一定量の修正を促進することが有効である. また修正量が減少したときに, 教員の原稿確認時に学生の原稿を点検・添削することで原稿の完成を早めることができる.

式 (2) のような修正スキルの学生は, 具体的な指摘をせずに原稿の確認のみで学生の創造的解決や創造的活動を認めることによって原稿の作成を促進できる可能性を示唆している. 他方, 式 (4) と式 (5) のような修正スキルの学生に対しては, 自己推敲のポイントを示す指導スタイルの方が考える時間を提供することができるので有効である. 具体的には, 言語情報とともに絵や図などを掲示する, または考える時間を提供することでエラー率が高いときの理解度を補助することが重要となる.

7 終わりに

本稿では, 複数の修正スキルを想定し, 原稿の完成時間に与える影響について検討をした. また, それに対して改善策の提案をおこなった. 今後の課題として, 原稿のセクションごとに学生の修正スキルを考慮にされることなどが挙げられる.

参考文献

- [1] 木村光宏, “2 段階テストに基づくソフトウェア信頼性評価法について-静的モデルと動的モデルの併用-”, 日本オペレーションズ・リサーチ学会秋季研究発表会アブストラクト集 2012, pp. 94-95, 2012/9. [2] 小山透, 『科学技術系のライティング技法』, 慶應義塾大学出版会株式会社, 2011. [3] 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. 1994. [4] Mesquite Software, <http://www.mesquite.com>. [5] 日本教育工学会, 『教育工学事典』, 実教出版株式会社, 2000.