

学習者特性を考慮した大学等における 原稿作成指導手法の検討 -ソフトウェア信頼度成長モデルによる-

土井 崇^{†1} 奥田 隆史^{†1} 井手口 哲夫^{†1} 田 学軍^{†1}

概要：経済産業省では「職場や地域社会で多様な人々と仕事をしていくために必要な基礎的な力」を「社会人基礎力」として定義している。社会人基礎力は、受動的に講義を受講するだけでは身に付けることが難しい能力である。そこで、本研究ではこの能力を向上する機会として、理工系学生が教員等と共著の学会原稿の作成を行なっていることに注目する。この原稿作成プロセスを学生の能動的学習を促すアクティブ・ラーニングをおこなう機会と捉えることで学生の能力向上を支援するとする。この原稿作成プロセスをアクティブ・ラーニングと捉えた場合の問題点は、学生の原稿作成・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、本稿では原稿作成をソフトウェア信頼度成長モデルにより表現することで、学生に対しての原稿作成指導手法について検討する。また、学生の原稿作成について不完全な修正も考慮することで、より現実的なモデルとした。

Performance Evaluation of Academic-Document Production Process using Software Reliability Models

TAKASHI DOI^{†1} TAKASHI OKUDA^{†1} TETSUO IDEGUCHI^{†1} XUEJUN TIAN^{†1}

1. はじめに

近年、企業や若者を取り巻く経済環境の変化により、「基礎学力」や「専門知識」に加え、それらをうまく活用していくための「社会人基礎力」を大学教育において意識的に教育することが重要となっている [1]。ここで社会人基礎力とは、経済産業省が 2006 年から提唱している「職場や地域社会で多様な人々と仕事をしていくために必要な基礎的な力」のことであり 3 つの能力（「前に踏み出す力」、「考え抜く力」、「チームで働く力」）からなる。

この社会人基礎力は、具体的な講義を受講するだけで身につく能力ではないため、あらゆる講義においてアクティブ・ラーニング（課題研究やディスカッション、プレゼンテーションなど学生の能動的な学習） [2] の機会を与えることが重要となる。また、効率的にアクティブ・ラーニング

を進めることを考察することも必要である。

さて、アクティブ・ラーニングという名称は使われていなかったものの、理工系学部の卒業研究や修士研究では、理工系学生が教員等と共著の学会原稿作成を頻繁に行っている。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が生まれる。つまり、この原稿作成プロセスを学生がアクティブ・ラーニングをおこなう機会と捉えると、教育の成果として単なる研究論文だけでなく、社会人基礎力をも教育していると考えられる。よって、教員等と共著の学会原稿作成する原稿作成プロセスは、社会人基礎力が求めている 3 つの能力全てを活用するプロセスでもある。

なお、原稿作成のような限られた時間の中で目的を達成することが必要な場合、教員が一方的に指導をおこなう方が時間が掛からないことが多い。しかし、学生に知識の伝達が行われなければ、能力の向上は望めない。体験を通し

^{†1} 現在、愛知県立大学大学院 情報科学研究科 情報システム専攻
Presently with Graduate School of Information Science and
Technology, Aichi Prefectural University

て発見した問題の解を求めて能動的に理論を学ぼうとすることで、実体験が確実に知識として学生の中に定着していく。

ただし、原稿作成プロセスを学生の能力向上を目的とするアクティブ・ラーニングをおこなう機会として捉えたときの問題点は、プロセス終了時間が、当該学生の原稿作成・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・修正間隔と学生の原稿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を応用した確率モデルを用いて検討する。本稿では、学生の原稿作成について不完全な修正も考慮することで、より現実的なモデルとする。

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、学生の原稿修正についてのモデルを議論する。5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修正スキルの確率モデルを提案する。次に、6節において、学生の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。7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8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 [4] 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に紙や電子ファイルの形式で教員に提出し、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原稿作成・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、関連する資料を学生に提供する場合があるとする。学生の原稿作成・修正スキルは、想定原稿の構成セクションにより異なる場合もあるが本稿では、構成セクションによらず、1つの原稿すべて同一の原稿作成・修正スキルであるとする。

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。

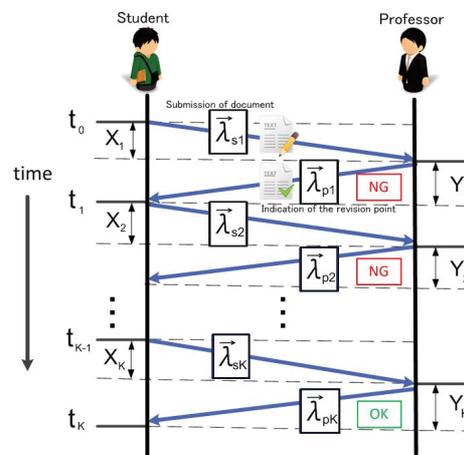


図1 原稿作成プロセスモデル

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。

- (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 回目の学生の原稿作成時間 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こなう。また、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. デバッグ環境

本研究では想定原稿を、文章構成が定型かつ明確に構成されることから、自然言語の中でも形式言語に近い性質があるとし、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、ソフトウェア信頼度成長モデル [5] を利用して表現する。本論文で想定する原稿のエラーには、次の2種類があるものと仮定する。

- F1** 原稿を書き始めた当初から含まれている原稿のエラー
- F2** 原稿を修正する際にその修正に伴い発生した原稿のエラー

また、教員が指摘するエラーについては、F1とF2のいずれかであるか区別はできないものとする。F1のエラー発生率は、原稿作成時間 t とすると $f(t)$ として表す。また、F2のエラー発生率は、その発生は原稿作成時間に関してランダムであり、一定の $\alpha(\alpha > 0)$ により表す。本研究では、エラーが発生する環境として完全デバッグ環境と不完全デバッグ環境を想定する。完全デバッグ環境とは、学生作成原稿においてF2のエラーが発生せずF1のエラーのみが発生する環境である。また、不完全デバッグ環境とは、学生作成原稿においてF1とF2の両方のエラーが発生する環境である。

4.1 完全・不完全デバッグ環境

完全デバッグ環境においては修正において新たなエラーが発生しない。しかし、一般的に原稿を修正する場合など、修正により予期せぬ新たなエラーが発生する不完全デバッグ環境である。したがって、F1およびF2のソフトウェア故障の発生現象を同時に考えるとき、原稿作成時間 t における原稿エラー発生率は

$$h(t) = \alpha + f(t) \quad (1)$$

により与えられる。式(1)より、時間区間 $(0, t]$ において発見される総期待エラー数は

$$\left. \begin{aligned} H(t) &= \alpha t + F(t) \\ F(t) &= \int_0^t h(x) dx \quad (t \geq 0) \end{aligned} \right\} \quad (2)$$

となる。そこで、本原稿では式(1)の $h(t)$ を強度関数、式(2)の $H(t)$ を平均値関数とするNHPP(nonhomogeneous Poisson process)モデル

$$\begin{aligned} Pr\{N(t) = n\} &= \frac{\{H(t)\}^n}{n!} e^{-H(t)} \quad (n = 0, 1, 2, \dots) \quad (3) \\ H(t) &= \int_0^t h(x) dx \quad (t \geq 0) \quad (4) \end{aligned}$$

に基づくソフトウェア信頼度成長モデルを議論する[6]。ここで、 $N(t)$ は時間区間 $(0, t]$ において発生する総期待原稿エラー数を表す確率変数であり、 $Pr\{\cdot\}$ は確率を表す。式(3)および式(4)で与えられるNHPPモデルの強度関数 $h(t)$ は、ランダムに起こる現象を記述するのに有用な同次ポアソン過程(homogeneous Poisson process)の強度関数 α (一定)を $f(t)$ に合成したものとなっている。

5. 原稿のエラー率と学生の修正スキル

ここで、学生が教員に原稿を i 回目の提出をした際の原稿のエラー率ベクトル $\vec{\lambda}_{si}$ を、

$$\vec{\lambda}_{si} = \{\lambda_H, \lambda_{Ab}, \lambda_{In}, \lambda_{Mi}, \lambda_{Co}\} \quad (5)$$

とする。なお、想定原稿の5つのセクションごとのエラー

率はそれぞれ独立している。また、原稿の各セクションの初期エラー数 a と、修正によって新たに発生したエラー数 αt 、式(3)の確率変数 $N(t)$ より、原稿の各セクションのエラー率 λ は、

$$\lambda = 1 - \frac{N(t)}{a + \alpha t} \quad (6)$$

となる。本稿では式(2)における $H(t)$ を学生の修正スキルとし、完全デバッグ環境($\alpha = 0$)と不完全デバッグ環境($\alpha > 0$)の両環境において、以下の(7)~(11)を考える。

$$H(t_i) = \alpha t_i + a - b(a - N(t_{i-1})) \quad (7)$$

$$H(t) = \alpha t + \frac{a}{c} \quad (8)$$

$$H(t) = \alpha t + a(1 - e^{-dt}) \quad (9)$$

$$H(t) = \alpha t + \frac{a}{1 + e^{-p(t-r)}} \quad (10)$$

$$H(t) = \alpha t + a \cdot q^{(e^{-p(t-r)})} \quad (11)$$

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(7)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(8)は学生の修正数が一定数の場合、式(9)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(10)と式(11)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。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。

6. 数値例

学生の修正スキルをシミュレーションにより評価する。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[7]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、完全デバッグ環境において75日で原稿エラーが基準値以下になるように6.1節と6.2節の各節でパラメータを定める。

6.1 完全デバッグ環境

完全デバッグ環境($\alpha = 0$)において、各修正スキルに対する平均エラー率との関係を図2に示す。各パラメータの値は $\mu_s^{-1} = 4[\text{day}]$ 、 $\mu_p^{-1} = 1[\text{day}]$ 、 $a = 100$ 、 $b = 0.75$ 、 $c = 60$ 、 $d = 0.08$ 、 $p = 0.15$ 、 $q = 0.5$ 、 $r = 30$ と定めた。

6.2 不完全デバッグ環境

不完全デバッグ環境($\alpha = 0.25$ 、 $\alpha = 0.5$)において、各修正スキルに対する平均エラー率との関係をそれぞれ図3と図4に示す。各パラメータの値は、完全デバッグ環境と不完全デバッグ環境において比較するため6.1節と同様とした。

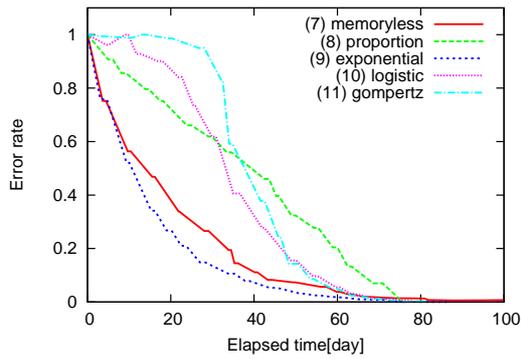


図 2 完全デバッグ環境における各修正スキルの平均エラー率の推移

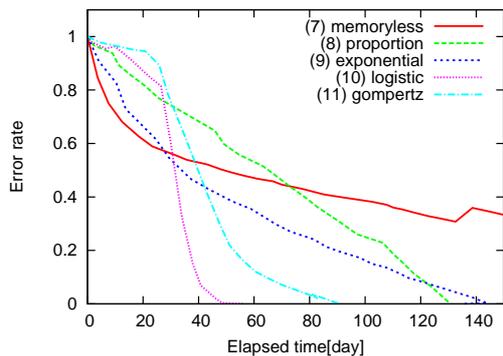


図 3 不完全デバッグ環境 ($\alpha = 0.5$) における各修正スキルの平均エラー率の推移

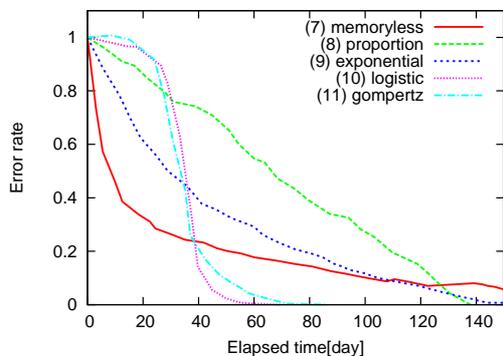


図 4 不完全デバッグ環境 ($\alpha = 0.25$) における各修正スキルの平均エラー率の推移

図 3 と図 4 より、不完全デバッグ環境において大きく影響を受ける修正スキルは、式 (7)～(9) の修正スキルである。また、式 (7) においては前回のエラー数にのみ依存して変化するため、不完全デバッグ環境においてはエラー率が一定以下にならない可能性もある。

7. 検討・考察：原稿作成指導への活用

大学におけるアクティブ・ラーニングを考慮した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 [8][9][10]。数値例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。完全デ

バッグ環境と不完全デバッグ環境におけるフィードバック例をそれぞれ 7.1 節と 7.2 節に示す。

7.1 完全デバッグ環境における指導例

完全デバッグ環境における指導例は、それぞれの修正スキルの特徴において検討する。式 (7) と式 (9) 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 (8) 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 (10) と式 (11) 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。

7.2 不完全デバッグ環境における指導例

不完全デバッグ環境における指導例は、それぞれの修正スキルの特徴と新たなエラーの発生についてを検討する。

式 (7) のような修正をする学生に対しては、前回のエラーに依存して変化するため不完全デバッグ環境においては終了しない可能性がある。そのため、このような修正スキルを持つ学生は修正する際に新しくエラーが発生しないように指導する必要がある。ソフトウェア開発手法の 1 つに、1 人がコードを書き、もう 1 人がそれをチェックしながらナビゲートするペアプログラミングと呼ばれる手法がある。そのことより、原稿も同様に 1 人が作成しもう 1 人がチェックしながらナビゲートする手法が有効である。

式 (9) のような修正スキルの学生は、修正する際に新しくエラーが発生する割合が高くと、エラー率の低下が一定量になる。しかし、原稿作成プロセスを開始した直後は他の修正スキルよりもエラー率の低下量が高いことから、現在の状態を明確に把握しつつ、過去のフィードバックを迅速に反省させるような心構えを指導することが重要である。

式 (8)～(11) のような修正をする学生は、完全デバッグ環境と近い性質を持つため完全デバッグ環境と同様の指導法が有効である。

8. おわりに

本稿では、学生が原稿作成をおこなう機会をアクティブ・ラーニングとして捉え、学生の能力による原稿作成指導手法についてソフトウェア信頼性モデルを応用した確率モデ

ルを用いて検討をおこなった。また、より現実的にするために完全デバッグ環境と不完全デバッグ環境を考慮し、原稿作成指導手法についての提案・検討をした。

今後の課題としては、学生が同時期に原稿作成をおこなうことを考慮し、学生同士の相互の影響を検討するなどが挙げられる。

謝 辞

本研究をまとめるにあたり、貴重なご助言をいただいた岐阜市立女子短期大学木村充位博士に深く感謝いたします。

参考文献

- [1] 「社会人基礎力」とは - 経済産業省, 入手先 (http://www.meti.go.jp/policy/kisoryoku/kisoryoku_image.pdf) (2013.04.21).
- [2] 2011 年度 大学のアクティブラーニング調査報告書 (質問紙調査報告), 入手先 (http://www.kawaijuku.jp/research/file/2011_houkokusyo.pdf) (2013.04.21).
- [3] 「ICT を活用したアクティブラーニング」報告書, 入手先 (<http://www.kals.c.u-tokyo.ac.jp/gp/archives/InternationalSymposium2008.pdf>) (2013.05.08).
- [4] 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.
- [5] 山田茂, 三木貴史 “潜入フォールトを考慮した不完全デバッグモデルと適合性評価”, 情報処理学会論文誌, vol. 39, No. 1, pp. 102-109, 1998.
- [6] 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.
- [7] Mesquite Software, <http://www.mesquite.com> (2013.04.09).
- [8] 日本教育工学会, 『教育工学事典』, 実教出版株式会社, 2000.
- [9] 荒井一博, 『教育の経済学 -大学進学行動の分析-』, 有斐閣, 1995.
- [10] アマルゴン, 『iTunes U と大学教育 Apple は教育をどのように変えるのか?』, シナノ印刷株式会社, 2012.
- [11] 大学におけるアクティブラーニング調査報告書, 入手先 (http://www.sanno.ac.jp/exam/career/pdf/active/report_activelearning.pdf) (2013.05.15).

付 録

A.1 大学教育におけるアクティブ・ラーニング

ここでは、大学教育におけるアクティブ・ラーニングの重要性について説明する [11].

アクティブ・ラーニングとは、「能動的な学習」のことで、教員が一方的に知識伝達をする講義スタイルではなく、課題研究やPBL (project/problem based learning), ディスカッション, プレゼンテーションなど学生の能動的な学習を取り込んだ授業を総称する用語である。よって、アクティブ・ラーニングが示す授業の形態や内容は非常に広く、その目的も大学や学部・学科によってさまざまである。

アクティブ・ラーニングは、説明概念としては新しいも

のであるかもしれないが、教育手法そのものは古くから大学教育の中に一部では組み入れられている。アクティブ・ラーニングが今、あらためて注目されるのは、世界的な大学教育の流れの中で「学習者中心の教育」の模索が本格化してきたこととも呼応する。そして「学習者中心の教育」すなわち「教員が何を教えたか」ではなく「学生が何をできるようになったのか」を基準として教育を教える場合、講義形式の授業だけではなく学生が能動的に授業に参加する授業形態が今まで以上に求められることとなる。