

仕様書の品質向上のための作成時の視線追尾によるユーザウェアネス算出 User-awareness from Eye Line Tracing during Specification Writing to Improve Specification Quality

若竹 慶則[†]
Wakatake Yoshinori

原田 史子[‡]
Harada Fumiko

島川 博光[†]
Shimakawa Hiromitsu

1. はじめに

ソフトウェア開発において、上流工程における仕様書や設計書の品質が下流工程の成果物の品質に影響を及ぼす。しかし、ソフトウェア開発の途中段階やリリース後に検出される欠陥の中に、ソフトウェア仕様書や試験仕様書での記述漏れが原因である不具合が存在するため、当該仕様書の品質を担保するレビューが存在する。しかし、人手によるレビューにて検知しているため、人的負荷が高い。また、レビューそのものの品質は、個人の能力差が大きく [1]、同一人物であっても、時間的な制約や集中度合いなどにより、品質が異なることも問題となっている。

近年、文書を読んでいる際のコンテキストの理解度を視線情報から推定する研究 [2][3] に代表されるように、視線情報を使用したコンテキストの分析が試みられている。視線情報を応用した先行研究として、コンテキストを読み流しているか否かリアルタイムで判別する研究がある [4]。しかし、仕様書の品質の評価には、流し読みの有無だけでは不十分で、注意力や集中力、理解度などが影響すると考えられる。

そこで、本研究では、エンドユーザの振る舞いを思い描きながら仕様書を作成すれば、仕様漏れ・試験漏れは大幅に減るであろうと期待される。これができる記述者とそうでない記述者では視線の動きに違いが出るのではないかと考え、本研究は視線計測情報を分析する。基本設計に値するソフトウェア仕様書（以下、SW 仕様書）から、結合テストに値する試験仕様書を作成するさいの記述者の視線情報を用いて、仕様作成者の User-awareness の度合いを推定する手法を提案する。そして、SW 仕様書、試験仕様書のどこをテキストマイニングの対象領域とすべきかを判別する。本手法を用いることで、品質の悪い仕様書が特定でき、仕様漏れ・試験漏れを防ぎ、高品質なソフトウェア開発を実現させる。

2. 仕様書作成時の内容把握と業務効率化

2.1 仕様書間の不整合検出の現状

TIAN 氏らは、word2vec を用いて単語の分散表現を生成し、文章に存在する単語の総和の平均を文章の特徴量として利用している [5][6]。しかし、単語に焦点を当てているため、文章の順序構造については考慮できていない。また、同一単語に着目しているため、試験仕様書に存在する詳細な試験項目間との対応を図ることは難しい。本研究では、SW 仕様書に規定された設計項目と詳細な試験項目との対応を図るためには、単語に焦点を置いたテキストマイニングだけではハードルが高いと考え、センサーを用いて仕様書のどこをテキストマイニングの対象領域とすべきかを判断する。

Egyed らや Reder らは、UML で表記された設計モデル間での不整合を検出して、修正する方法を提案している [7][8]。ただし、その方法は UML で表記された成果物のみを対象としている。仕様書には、UML でなく自然言語で記述されたものが多い。よって、仕様書の設計項目と試験項目の対応を網羅的に検証することは困難である。本研究では、自然言語で記述された仕様書を対象とする。

3. 視線情報からの User-awareness の推定

3.1 User-awareness

User-awareness とは、特定のコンテキストにおいてエンドユーザがどのように振る舞うかを想定することである。本研究では、SW 仕様書の作成者が想定していない事柄を、試験仕様書の作成者が想定することに当たる。記述時に User-awareness をもつ人（適切に仕様書を作成している人）とそうでない人を推定できれば、品質の悪い仕様書が特定でき、仕様書の品質向上に寄与できる。仕様書作成時の User-awareness を推定する手法を図 1 に示す。

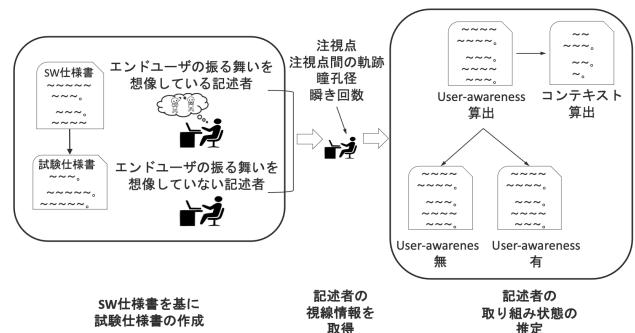


図 1: User-awareness 推定手法

[†]立命館大学

[‡](株) コネクトドット

3.2 User-awareness とコンテキスト推定時の動作

SW 仕様書から試験仕様書を作成するさい、取り組み状態を、以下の 2 つに分ける。

- (1) 仕様書の内容・構造を理解しようとしている。
- (2) エンドユーザ視点で想像している。

上記 2 つの取り組み状態は、以下のように視線情報と関連すると仮定する。

(A) (1) では、SW 仕様書内の特定のキーワードを注視している割合が大きくなる。また、仕様書間・仕様書内の構造（節番号）間を視線先が飛び交う。また、暗算や連想などをするさいには内的な注意が活性化されるので、単位時間当たりの瞬き回数が増加する [9]。

(B) (2) では、モニタ外を注視している割合が大きくなる。また、(A) と同様に単位時間当たりの瞬き回数が増加する。

上記の仮説を支持する結果を得ることができれば、仕様書における重要単語を割り出せ、テキスト分析の対象となる箇所を絞り込むことができる。また、注視点先の対応から仕様書間・仕様書内の構造(節番号)間の対応が明らかになり、テキストマイニングの対象領域を限定できる。よって、無駄なレビューを避けることに繋がる。また、User-awarenessの有無から記述者へ改善点を指摘することが可能となり、仕様書の質の向上が期待できる。

視線情報から User-awareness をもつ仕様書とそうでない仕様書を手早く判別できれば、これら2つのグループの多数の例を手に入れることができる。多数の例が得られれば、これら2つのグループのそれぞれに属する仕様書に共通するテキスト上の特徴が高い精度で同定できる。いったんこのような特徴が把握できれば、それ以降は、テキスト分析で新しい仕様書の特徴がどちらのグループに類似しているかを判断することにより、設計項目と試験項目との間での記述漏れが起こりやすいSW仕様書と試験仕様書の組を摘発することができる。

3.3 視線情報による特徴量の抽出

アイトラッカを用いて仕様書作成時の視線情報を取得し、取得した視線情報を基にして1頁単位で説明変数を算出する。アイトラッカは視線情報を計測する専用のデバイスであり、本実験ではトビー・テクノロジー株式会社の Tobii Pro ナノを用いる。

アイトラッカで取得できる四つの基本情報(注視点: 1箇所を注視している視線の集まり, 跳躍: 注視点間の目の軌跡, 瞳孔径: 瞳孔の大きさ, 瞬き: 瞬きの有無)を基に、計18の説明変数を算出する。

英語の理解度と視線の関連を調査した研究[4]では、特定の視線情報が理解度と関連した。そこで、Random forestによる機械学習アルゴリズムを用いて、User-awareness算出に影響を及ぼしている説明変数を抽出する。Random forestは各説明変数の重要度も同時に算出するので、仕様書の品質への重要度が低い説明変数は削除できる。

4. 仮説の検証実験

仮説を検証するため、仕様書作成を模した実験を行い、実験中の視線情報を計測し、視線と仕様書精度の関係を分析する。

検証実験に使用するSW仕様書は、大学のプログラミング演習で使用されたオセロゲーム開発課題のレポート問題文を基にして、開始時の初期表示/盤面への石の配置/挟み撃ちの3項目の構成に改変し、計10頁とする。

被験者に対して、SW仕様書と試験仕様書をモニタに提示し、試験仕様書作成時の視線をアイトラッカで計測する。まず、事前知識のみでSW仕様書から試験仕様書を作成する。次にSW仕様書を試験するための適切なテスト観点を被験者に伝え、同一のSW仕様書から試験仕様書を作成する。テスト観点を伝える前後での視線情報の違いから動作特徴を検出できる。実験終了後にアンケートを実施し、被験者特性として、実験時の集中度や対象コンテキストへの理解度を取得する。

5. おわりに

本紙では、User-awarenessを算出する人、あるいは、User-awarenessを算出するさいには、その仕様書作成の仕方に共通点があると考え、視線情報を用いて仕様書作成者のUser-awarenessの度合いを推定する手法を提案した。今後は複数人に対して実験し、仮説を検証する。

参考文献

- [1] B. Nuseibeh, J. Kramer, and A. Finkelstein. Analyzing individual performance of source code review using reviewers' eye movement. No. 8, p. 133-140, 2006.
- [2] 和代吉村, 仁史川市, Kai Kunze, 浩一黄瀬. アイトラッカで取得した視点情報と文書理解度の関係(パターン認識・メディア理解). 電子情報通信学会技術研究報告: 信学技報, Vol. 112, No. 495, pp. 261-266, mar 2013.
- [3] Ayano Okoso, Takumi Toyama, Kai Kunze, Joachim Folz, Marcus Liwicki, and Koichi Kise. Towards extraction of subjective reading incomprehension: Analysis of eye gaze features. p. 1325-1330. Association for Computing Machinery, 2015.
- [4] Ralf Biedert, Jörn Hees, Andreas Dengel, and Georg Buscher. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, p. 123-130. Association for Computing Machinery, 2012.
- [5] Qingsong Tian, Qinghua Cao, and Qing Sun. Adapting word embeddings to traceability recovery. In *2018 International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pp. 255-261, 2018.
- [6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
- [7] Alexander Egyed, Emmanuel Letier, and Anthony Finkelstein. Generating and evaluating choices for fixing inconsistencies in uml design models. *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, Vol. 0, pp. 99-108, 09 2008.
- [8] Alexander Reder and Alexander Egyed. Computing repair trees for resolving inconsistencies in design models. New York, NY, USA, 2012. Association for Computing Machinery.
- [9] Pauline Cho, Caren Sheng, Carol Chan, Raymond Lee, and Joey Tam. Baseline blink rates and the effect of visual task difficulty and position of gaze. *Current eye research*, Vol. 20, pp. 64-70, 01 2000.