

平成22年度 情報工学コース卒業研究報告要旨

阿草 研究室	氏 名	高 井 康 勢
卒業研究題目	コーディング規約違反に基づくソフトウェアメトリクス	
<p>ソフトウェア開発の現場では、ソフトウェアやプロジェクトの規模や品質等を測定するために、ソフトウェアメトリクスが使われている。ソフトウェアメトリクスを導入することで、ソフトウェア開発を管理できるようになり、効率的かつ計画的にソフトウェアの生産を行うことが可能になる。</p> <p>しかし、現在までに提案されてきたソフトウェアメトリクスには、開発途中の成果物や開発者の状態を測定できないという問題がある。</p> <p>本研究では、開発途中の成果物や開発者の状態を測定するためのソフトウェアメトリクスの提案にあたり、コーディング規約に着目した。コーディング規約は、コーディングを行う上で守るべきルールである。コーディング規約を利用する目的とは、ソースコードの可読性や保守性を向上させ、将来発生する可能性のある不具合を予防するためである。</p> <p>コーディング規約をソフトウェアメトリクスとして利用する理由は三つある。一つ目は、コーディング規約は既に普及しているため、ソフトウェアメトリクスの導入が容易であるためである。二つ目は、コーディング規約違反が発生する要因が多岐に渡っているため、ソフトウェアの様々な状態を評価できるためである。三つ目は、コーディング規約は、将来の不具合を予防できるので、将来のソフトウェアの状態を予測できる可能性があるためである。</p> <p>本研究では、コーディング規約違反に基づくソフトウェアメトリクスを提案する。本ソフトウェアメトリクスでは、行あたりのコーディング規約違反数 (NVPL) や、コーディング規約違反が発生してから修正されるまでの期間の長さの平均 (AVEP) を測定して利用する。また、本研究では実験できなかったが、本ソフトウェアメトリクスを利用することで評価できる可能性のあることについてまとめた。具体的には、プロジェクトの今後の状態を予測することや、プロダクトに不具合が存在する可能性の評価などである。</p> <p>コーディング規約違反をソフトウェアメトリクスとして利用するためには、検出したコーディング規約違反のログを取得する必要がある。本実験で用いた、CX-Checker というコーディング規約違反をチェックするツールには、コーディング規約違反のログを取得する機能がなかったため、ロギング機能を追加する拡張を行った。</p> <p>さらに本研究では、評価実験として、本研究で提案するソフトウェアメトリクスである NVPL が、機能追加と保守作業や機能削減の差を表現できることを確認するための実験を行った。リアルタイムカーネルである TOPPERS/ASP カーネルの複数のバージョンに対して、コーディング規約違反のチェックを行った。その結果、NVPL を用いることで、TOPPERS/ASP カーネルの開発ログの、機能追加と保守作業や機能削減を区別できた。このことにより、NVPL が、機能追加と保守作業や機能削減の違いを表現することを確認し、本研究の有効性を確認した。</p>		

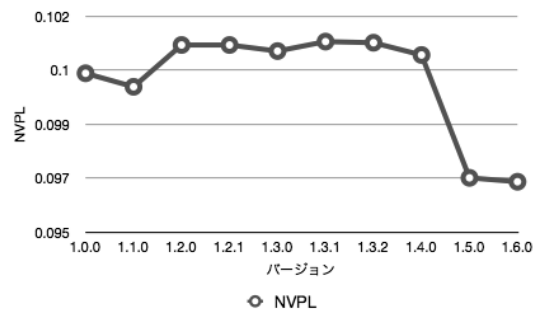


図:カーネルの各バージョンにおける NVPL の変化