

5ZC-2

## プログラム静的解析ツールを利用した ソフトウェア品質向上活動

田村朱麗 山田淳<sup>◎</sup> 小笠原秀人<sup>\*</sup>

株式会社 東芝

### 1. はじめに

ソフトウェアの開発は、開発場所の分散化が進み、規模もますます増大する傾向にある。このため、ソフトウェアの品質管理も分散開発環境下で実現することが要求される。また、ソフトウェアの品質を効率良く保証するためには品質管理ツールの導入が有効である。

本稿では、我々が試作したWebブラウザをインタフェースにした静的解析システムの概要を述べる。試作したシステムは、開発者が自由に利用できる環境と、プロジェクトの全ソースに対する品質情報を保存する環境を提供する。

### 2. 静的解析技術

高品質なソフトウェアを開発するためには、コーディングの初期段階から品質を意識することが必要だと思われる。プログラムの静的解析は、ソースコードの潜在的な不具合を開発工程の早い時期に指摘できる。人手によるコードレビューは、効果的である。しかし、効果的なレビュー速度は150行/hという報告[1]があり、規模の大きいソフトウェアの全ソースに対して実施するのは難しい。ツールによる静的解析の有効性は、大規模なソフトウェアであっても、全ソースプログラムに対して短時間に解析を行えることである。

しかし、特化した機能を持つツールを効果的に利用するためのノウハウを短期間に習得するのは難しい。静的解析ツールも例外ではない。そのため我々は開発部門に静的解析ツールの利用を技術サービスという形態で支援する活動を行っている。[2],[3]

### 3. 静的解析技術の効果的な利用方法

我々は、C言語のプログラム静的解析のためにQAC<sup>†</sup>というツールを利用している。

QACの機能のうち、我々は次の機能に注目している。  
コード解析：潜在的なソフトウェア欠陥の検出  
メトリクス計測：ソフトウェアの定量化  
そして、解析結果を有効に活用するために、(1)-(4)の活動を行ってきた。

#### (1) 警告メッセージの重要度分類

QACには、約800種類の警告メッセージを3つのランクに分類した。

表1 重要度ランク分類

ランク	特徴
A	重要な警告 (制御式での代入文の利用、型変換、未初期化、到達しない文等)
B	保守性、移植性に関する警告
C	スタイル等に関する警告

#### (2) 警告メッセージへの対応方法の提示

静的解析ツールによって指摘されるソースコードの例と、その対応方法を一覧表にした。

#### (3) ユーティリティツールの開発

次のようなユーティリティツールを開発した。

表2 ユーティリティツール

機能	ユーティリティツール
コード解析	重要度ランク指定解析
警告メッセージ	出現頻度・出現箇所一覧表
メトリクス	メトリクス値のグラフ化
プロセスデータ	時系列データの差分集計

これらの結果から、コードレビューやテストを重点的に実施すべき箇所を絞り込むことができる。

#### (4) Webでの情報提供

静的解析の動向や(1)-(3)に関する情報をイントネ

Support for Software Quality Improvement by Using Program Analysis Tools

<sup>◎</sup> Shurei Tamura, Atsushi Yamada: System Integration Technology Center, Information and Industrial Systems & Services Company, TOSHIBA Corporation, 3-22, Katamachi Fuchu-shi, Tokyo 183-8512 Japan

<sup>\*</sup> Hideto Ogasawara: System Engineering Laboratory, Research & Development Center, TOSHIBA Corporation 70 Yanagi-cho Saiwai-ku Kawasaki-shi Kanagawa 210 Japan

<sup>◎\*</sup> {shurei.tamura, atsu.yamada, hideto.ogasawara}@toshiba.co.jp

ットで提供することにより、ノウハウを共有する。このような支援の結果、自主的に運用をする部門も増えつつある。これらの部門では、解析を実施する担当者をアサインし、単体テスト後、結合テスト前等のタイミングでプロジェクトの全ソースを対象に行っているケースが多い。しかし、各開発担当者が利用するには至っていないのが現状である。そこで、開発担当者が静的解析ツールによる品質評価を実施しやすい環境を提供する、Web ブラウザをインターフェースにしたシステムを試作した。

#### 4. Web を利用した品質解析システム

Web を利用したイントラネット形品質解析システムを試作した。試作したシステムは、ユーティリティツールと同様の機能を Web ブラウザから実行できる。システムを Web アプリケーション形式で提供する利点は、クライアント側にインストール作業が発生しないこと、操作性が良いことである。そのため、導入時及び運用時の負荷が大幅に軽減する。また、手軽に利用できることにより、解析サイクルを短縮できる。

##### 開発担当者用の環境:

提供機能: 解析機能

利用方法: 各自が開発したソースを評価

##### プロジェクトの品質担当者用の環境:

提供機能: 解析機能, 品質データ保存機能

利用方法: 単体テスト前, 結合テスト前, 総合テスト前などのタイミングで全ソースを評価しその結果を保存する。また、各タイミングでの結果を比較し、プロジェクト全体の品質が把握する。

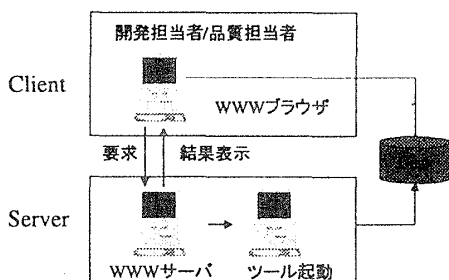


図 1 動作環境

#### 5. 適用事例

デバック終了時点で1回目の静的解析し、約10日後に2回目の解析を実施した。評価の対象は、ランク

Aの警告である。

表 3 適用事例

	1回目	2回目
ファイル数	134	138
モジュール数	3022	3072
コード行数	167.9K	171.4K
警告数	510	340
警告密度	3.0/K	2.0/K

2回目の静的解析では、ソースコードの行数は3.5K増加しているが、出現した警告数は減少している。また、1K当りの警告の出現頻度は、3件から2件に減少している。2回目に出現した340件の警告のうち、新たに出現した警告は24件であり、残りの316件は修正の必要がないと判断されたものである。つまり、この間に総出現数の36%に当たる184件の警告が修正された。このように、静的解析を実施することにより短期間でソースコードの品質が向上する。このことから、コーディングのできるだけ早い時期から静的解析を利用することがより効果的であると考えられる。

#### 6. まとめ

本稿では、「プログラム静的解析を利用したソフトウェア品質向上支援活動」について述べた。第3者の立場で静的解析技術の支援活動をしてきた結果、静的解析ツールを導入する部門が増え、節目毎に解析が実施されるようになった。しかし、開発者が各自のソースコードに対して利用している場面は多くない。我々は、この問題を解決するためにWebブラウザをインタフェースにした静的解析システムを提案した。

#### 参考文献

- [1] Thomas Liedtke 他: On The Benefits of Reinforcing Code Inspection Activities Process Improvements in the Telecomm Environment, EuroSTAR'95, 1995
- [2] 會澤他: "プログラム解析・計測ツールを利用したソフトウェア品質向上支援活動" 日課儀連, 第17回 ソフトウェア生産における品質管理論文集, 1997.
- [3] 小笠原他: "プログラム静的解析の適用事例と効果分析", ソフトウェア・シンポジウム'98, 1998

<sup>1</sup> 英国 PRL(Programming Research Ltd)社 で開発。