トップエスイー修了制作

大規模ソフトウェアにモデル検査を適用するための抽象化指標検討

富士通株式会社　安岡大知　yasuoka.hirotom@jp.fujitsu.com

開発における問題点

モデル検査を実施するにあたって状態爆発という問題に直面する.
一般的に検査するモデルの抽象化によって状態爆発の回避が行なわれるが，粒度が粗すぎるとバグ検出出来なかったり，意図しない反例が検出される場合がある.

手法・ツールの適用による解決

通信制御ソフトウェアのバグ原因分析結果とソフトウェアメトリクス値をインプットとして，検証モデルを適切な粒度に保つ抽象化指標を確立し，抽象化判断に活用することで問題を解決する.

抽象化指標の確立

■ バグ原因分析

原因の大半は状態遷移中のイベント競合受信や異常発生を契機に状態切替えを行った際のデータ操作誤り.

＜バグ原因例＞

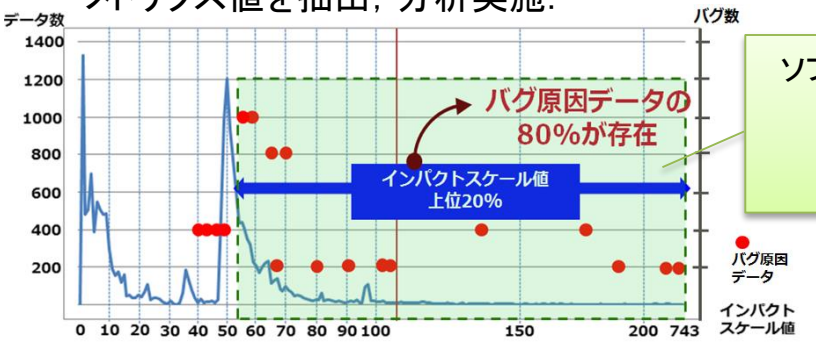
状態AからB遷移中に状態Aに戻る

状態	データ1	データ2	データ3
A	ON	ON	ON
↓			
B	ON	ON	OFF
↓			
A	ON	ON	OFF

同じ状態Aでデータ内容が異なる

■ データに着目した抽象化指標値

各データのソフトウェアメトリクス値とバグ発生データのソフトウェアメトリクス値を抽出，分析実施.



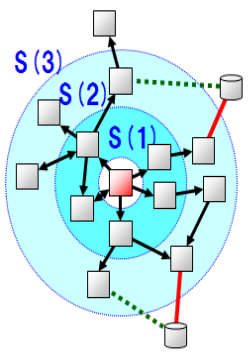
ソフトウェア全体のデータ数
→約18000
上位20%のデータ数
→約3600

上位20%のデータをモデル検査対象，それ以外を抽象化対象とすることで検証モデルを適切な粒度に保つことが可能と考える.

採用メトリクス

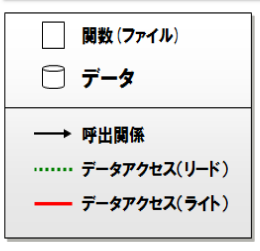
インパクトスケール®

プログラムから呼び出される別のプログラムや，参照・更新されるデータを辿っていき，それらの関係の強さで重み付けしながら計測することで，ソフトウェアの影響波及の範囲を表現するソフトウェアメトリクス



$$IS = S(1) + \alpha S(2) + \alpha^2 S(3) + \ldots$$

- S(d) … 最短距離dのプログラムとデータの個数
- α … 減衰係数（0 < α < 1）



情報処理学会論文誌
Vol.54 No.2 870–882 (Feb. 2013)

まとめ

2つの通信制御装置を対象に分析した結果を元に指標値検討を実施した.
今後，他装置のデータも対象に分析を継続して行なうことで，抽象化指標値の有効性を継続検証する.

＜今後の課題＞
現時点では，通信制御ソフトウェアのソースコードを対象にインパクトスケール値と混入バグとの関係について調査した結果にとどまっている.

今後，抽象化指標の活用・未活用それぞれのPromelaモデルを作成し，それぞれのモデルのバグ数の関係について分析を実施していく.