トップエスイー修了制作

形式仕様記述と実装間の整合性検証手法の構築

株式会社クレスコ　　黒田尭利　　t-kuroda@cresco.co.jp

開発における問題点

組み込みソフトウェア開発の上流工程に形式仕様記述を取り入れた場合に，下流工程に仕様を正確に反映できているかどうか，レビューなど人手だけで厳密に検証することは困難である．

手法・ツールの適用による解決

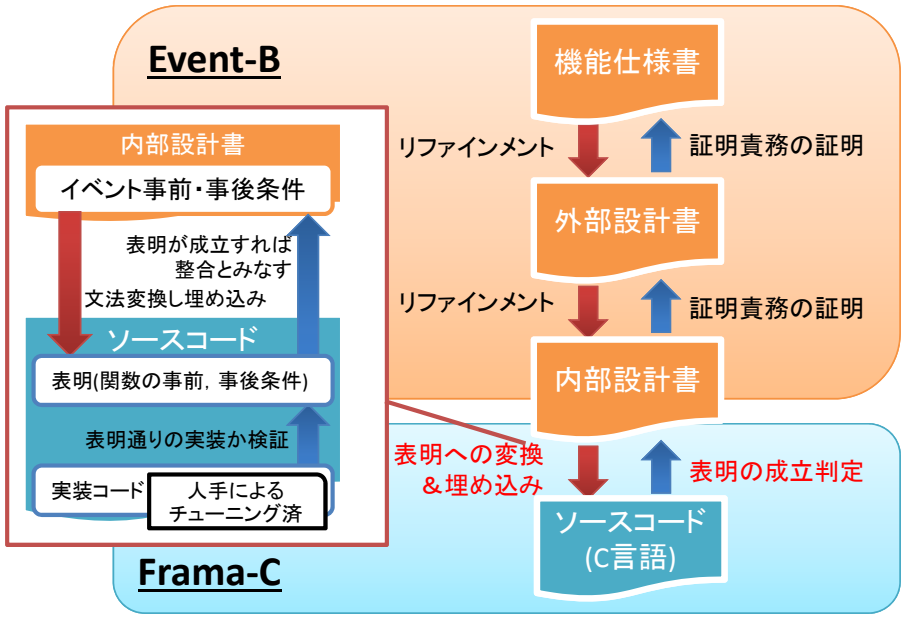
設計工程はEvent-Bのリファインメントの仕組みを利用，実装工程ではソースコード静的検証ツールであるFrama-Cを用いて，形式手法ツールを用いた厳密な整合性検証を行う．

手法の概要

2つのアプローチで整合性検証を行う．

- **「仕様書」と「設計書」間**
  - 既存の設計指針を踏襲し，Event-Bのリファインメントにより設計書を作成する．
  - ⇨**証明責務により整合性を評価する．**
- **「設計書」と「ソースコード」間**
  - 内部設計書からACSL表明*に変換，ソースコード中に埋め込む
    - チューニングや環境制約により人手で実装したソースコードを想定．
  - ⇨**実装に対するACSL表明の成立性を検証し整合性を評価する．**

*C言語用プログラムに対しあるべき振る舞い(仕様)を記述する言語．



評価結果

小規模なシステムの設計書、ソースコードを作成し，実開発時に発生しうる不具合を提案手法で検出できるか評価した．

- **仕様の実装漏れ**
  仕様に対する処理の欠落
  ⇨事後条件違反として検出された
- **実装時のチューニングによる不整合**
  仕様に対し過小な変数によるオーバーフロー
  ⇨事前条件違反として検出された

静的検証ツールの実行により，仕様と実装の不整合を検出できた．

まとめと課題

- **まとめ**
  次のアプローチで整合性の検証が実施できた．
  - Event-B設計書のリファインメント戦略
  - ACSL表明，C言語実装へのマッピングルール
- **今後の課題**
  - Event-B→ACSL変換・埋め込みのツール化
  - Event-Bによる設計過程，実装，ACSLのトレーサビリティ可視化