トップエスイー修了制作

制御ボードファームウェアの二重化制御機能を対象とした形式検証の適用

(株)富士通コンピュータテクノロジーズ　辻村 浩史　tsujimura.hirof@jp.fujitsu.com

開発における問題点

・二重化制御機能で用いる各種アルゴリズムは、個々のテストにより一定の品質を確保している。
・しかし、それらを実行するロール決定プロセスの並行制御で障害が発生してしまう。
・アルゴリズムのテストだけでなく、それを実行する並行プロセスについても、様々なケースをテストする必要がある。

手法・ツールの適用による解決

・VDM++により、プロセスの様々なケースごとに期待するMBの状態を仕様として定義した。
・定義した仕様を検査できるよう、様々なケースを再現可能なロール決定プロセスをモデル化した。

※ MB: 制御ボード

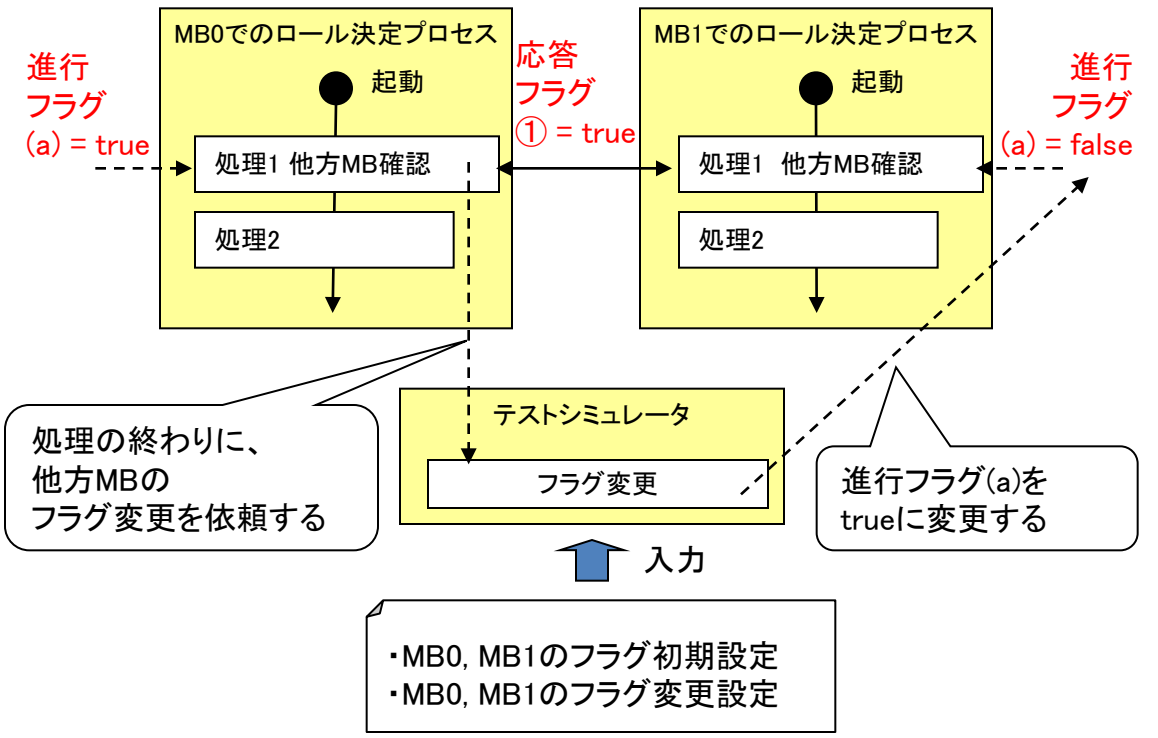
様々なケースを再現できるプロセスのモデル化

二重化制御機能では、2枚のMBがお互いにやりとりしながらロール決定プロセスを実行する

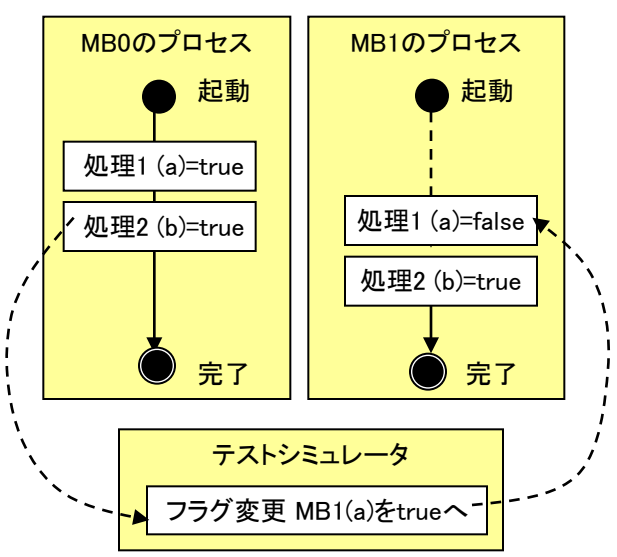
様々なケースごとに、MBが期待する状態となっているかを検査できる仕組みを考える

様々なケースを再現するため、以下を導入する

- 進行フラグ
  - ✓ trueの場合、処理を実行する
  - ✓ falseの場合、処理をストップする
- 応答フラグ
  - ✓ trueの場合、他方からの通信に応答する
  - ✓ falseの場合、応答しない
- テストシミュレータ
  - ✓ 各処理の終わりにフラグを変更する



プロセスの検査例



MB0のプロセスがMB1のプロセスを先行して完了するケース

MB0の処理2の終わりに、MB1の進行フラグ(a)を変更することで、再現することができる

このとき、各MBが期待する状態になっているかを検査する

MBに期待する状態の例:
MB0のロールがActiveで、パネルアクセス権を保持していること 等

評価

・プロセスの様々なケースごとに、各MBに期待する状態を仕様化できた

・更にVDM++で記述したことで、他の技術者が理解できる形で仕様を定義できた

・状態とフラグ、テストシミュレータを導入することで、様々なケースを再現、検査できるモデルを作成できた

・今回のテストシミュレータを実装言語で作成すれば、VDM++での検査を実装コードのテストとして実行できると考える