

形式手法による 組み込みソフトウェア開発

お待たせしました!

第10回
(最終回)

ソフトウェア・シミュレーションと
実ハードのテストを組み合わせる

藤倉 俊幸

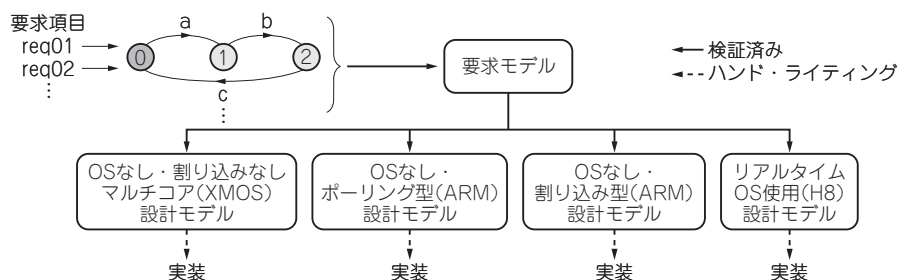


図1 今までの経緯と検証の考え方

「要求モデル⇒(ならば) 要求項目, 設計モデル⇒要求モデル, 実装⇒設計モデル」が成り立つなら, 「∴ (ゆえに) 実装⇒要求項目」と言える

長らく休載が続いていました本連載もついに最終回を迎えます。

本連載(2012年5月号から掲載)では、形式手法を利用した組み込みシステムの開発方法について説明してきました。カップラーメン・タイマの開発を例にとり、最初に要求項目の形式化を行い、次にそれらすべてを合成し要求モデルを生成しました(図1)。この要求モデルを使って、妥当なふるまいをするかどうかを調べました。そして、そのモデルに対して実装プラットフォームの制約を追加しつつ、動作単位に分割して設計モデルとしました。ここでの動作単位とは割り込みハンドラとメイン関数、あるいは、割り込みハンドラとタスク関数です。

この過程で、要求項目単位だったものをいったん一つに混ぜ込んで要求モデルとし、要求項目間の矛盾を検出しました。そして、いったんまとめたものを、それぞれのアーキテクチャで導出された動作単位に分割し、それを動作単位ごとの詳細仕様として実装を行いました。検証は、動作単位ごとに分割したものを合成して設計モデルとし、その設計モデルが要求モデルと矛盾しないかをLTSA (Labelled Transition System Analyser⁽¹⁾) というモデル検査ツールで確認しました。

いわゆる段階的詳細化と再合成を繰り返して検証してきたわけです。この方法は厳密ですが、何度も形式検証(分割・合成・検証)を行わなければならない、効率が悪く非現実的です。また、最終的にC言語を利用

する現状の実装方法では最後の段階で未検証の部分が残ってしまいます(コラムを参照)。そこで、HIL (Hardware-in-the-Loop) シミュレーションの考え方を導入してみます。

従来のテストではテスト・ケースと正解を用意してテストを実施し、テスト結果と正解が一致するかを確認しますが、HILの場合は正解モデルを使用してふるまいが一致するかを確認します。正解モデルは要求モデルから作成します。

HILとモデル・ベース開発

● コントローラとプラント・モデルで閉ループを構成

HILは、自動車や航空機分野で利用されてきた手法であり、最近ではエネルギー分野でも利用されています。これらの分野の特徴は、制御対象が微分方程式などで表される数学的モデルとなることです。そして、この数学モデルと制御理論を使ってシステムの効率や安定性を開発の初期の段階でシミュレートできることです。

この段階の一般的なモデリング・ツールは、MapleSimやMATLAB/Simulink, Dymolaです。また、システム・レベル設計では、AutoDeskなどの機械系の3D-CADモデルによって質量などのパラメータ設定や、電子系ではSPICEなどによって回路シミュ