

次世代宇宙機搭載ソフトウェア検証環境の研究

Study of spacecraft onboard software verification environment

上田 裕子[1]; 池田 茂[1]; 高橋 孝[1]

Hiroko, O Ueda[1]; Shigeru Ikeda[1]; Takashi Takahashi[1]

[1] 宇宙航空機構

[1] JAXA

ロケットや人工衛星など宇宙機の搭載ソフトウェア(Onboard Software; OBS)は、プロセッサ技術の進展に伴って大規模・複雑化が急速に進んでいる。一方、宇宙機の開発においては、社会的要請から期間短縮、低コスト化と並んで信頼性の向上が最重要課題である。そこで我々は、様々な宇宙機に共通に適用可能な基盤技術として搭載ソフトウェア検証環境を構築することを目指した研究を行っている。

搭載ソフトウェアの開発初期には、アルゴリズム設計検証のために搭載プロセッサや搭載リアルタイム OS など実行環境に依存しない数学モデルを用いてシミュレーションが行われる。搭載用コードが作成された段階では、コードを試験実行するために3種類の方式が用いられる。それらは、汎用計算機上で実行するフルソフトウェア・シミュレーション(FSS)方式、搭載プロセッサのソフトウェアエミュレータ(Instruction Set Simulator; ISS)上で搭載プロセッサ命令を逐次解釈して実行する方式、または、搭載プロセッサを組み込んだ評価ボードあるいは実際に搭載するハードウェア上で直接実行する方式である。それぞれ目的によって使い分けられるが、ソフトウェア・エミュレータ方式は、評価ボード等のハードウェアを用いた方式では実施困難なハードウェア異常・故障等の例外事象についてもきめ細かい検証が可能で、またソフトウェアの動作状況を詳細にモニタできることから、強力な検証ツールとなる。

いずれの方式にとっても搭載ソフトウェアの実行に必要な入力パラメータを供給し、出力パラメータを受け取るインターフェイスおよびシミュレータ等の環境が必要となる。シミュレータは、宇宙機によってそれぞれ異なるコンポーネントモデルと、コンポーネント間を関連付けデータを受け渡す機能および実行制御機能など、宇宙機に共通なフレームワークによって構成できる。

従って本研究は、個々の搭載ソフトウェアや宇宙機モデルの違いおよび検証方式による違いを吸収するインターフェイスと共通フレームワークを設計することが中心課題である。これまでにFSS方式とハードウェア方式を共通化するインターフェイスを試作した[1]。現在は、ハードウェア方式とソフトウェア・エミュレータ方式で可能な限り互換性の高いインターフェイス、およびシミュレータのフレームワークの検討を行っている。

[1]Takashi Takahashi, H. O. Ueda, N. Yoshioka, S. Hirano, M. Muranaka, and T. Kobori, Distributed Object-Oriented Framework for Spacecraft Simulations, 24th ISTS 2004-t-15.