

機器制御ソフトウェア向けテスト環境の一事例 — 構想 —

5C-7

鳥袋 潤† 日下 敏明†† 小泉 忍† 渡辺 哲也‡ 佐々木 保†

†(株)日立製作所 システム開発研究所 ††(株)日立超 LSI システムズ

‡(株)日立製作所 汎用コンピュータ事業部

1. はじめに

専用ハードウェアを用いるシステム開発では、ハードウェアとソフトウェアを並行して開発することが理想である。このためには、専用ハードウェアの完成以前にソフトウェアのテストを行なう必要がある。数人で開発する小規模ソフトウェアに対しては、従来からハードウェアの応答データを与えるソフトウェアモジュール(スタブ)を個々に作成するなど、ソフト開発者の属人的作業として行われてきた。

筆者らは複数種・複数台のコンピュータ機器・周辺機器を制御する機器制御システムにおいて、全体を制御するソフトウェア(単に制御ソフトとよぶ)のテスト方式・テスト環境を構築している。システムの高機能化に伴い、制御ソフトの比重が相対的に拡大し、開発規模・開発人員も増大している。この場合、上述の属人的作業の延長では対応が不可能である。

本稿では、制御ソフトのテスト環境に求められる要件の検討、従来のテスト環境の分類を行ない、開発構想を述べる。

2. 対象システムとテスト上の課題

機器制御システムは、オペレータからの操作に基づいて複数の制御対象装置を制御するとともに、制御結果・装置状態をオペレータに表示するものである。システム全体の制御は制御装置上で動作する制御ソフトで実現するが、制御対象装置との間には中継装置が存在する構成である(図 1)。

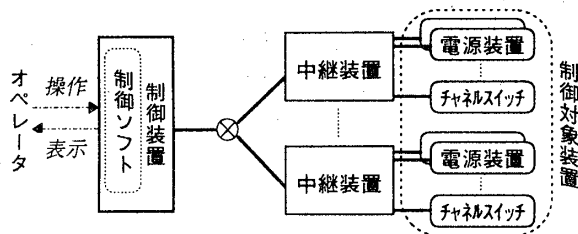


図 1 機器制御システムの構成

Case study on Test Environment for Hardware Control Software - Concepts

Jun Shimabukuro†, Toshiaki Kusaka††, Shinobu Koizumi†, Tetsuya Watanabe‡, and Tamotsu Sasaki‡.

† Systems Development Laboratory, Hitachi, Ltd.

†† Hitachi ULSI Systems Co., Ltd.

‡ General Purpose Computer Division, Hitachi, Ltd.

制御ソフトのデバッグ・テストを行なう上で、次の三点が課題となる。

(1) 中継装置完成以前でのデバッグ・テスト

制御ソフト全体での統合テスト時点では中継装置を並行開発中のため、制御対象装置を制御する機能の実機テストを行なえない

(2) システム構成バリエーションに応じたテスト

機器制御システムの納入前には実際の制御対象システム構成における動作確認(構成テスト)が必要であるが、構成するハードウェアを十分な期間確保することは困難である

(3) テスト準備工数の削減

開発期間短縮、製品の信頼性・品質の確保のため、上記(1)、(2)を行なうための準備に要する工数を従来に比較して削減する必要がある

上記の課題を解決するため、中継装置および制御対象装置の動作を模擬するテスト環境(以下では機器シミュレータとよぶ)の開発を構想した。求められる要件として、次の二点を設定した。

[要件1] 実際のシステム構成に応じ、機器シミュレータの構成実現・変更が容易であること

[要件2] テスト条件の指定/テストデータの作成が容易であること

3. 機器シミュレータのモデル

3.1 動作模擬方式の分類

一般にテスト環境(スタブ)における動作模擬実現方式は、次の三種類に大別できる。

(1) 完全模擬型

模擬対象の動作を完全に模擬し、自動的に応答する。応答内容が固定となるので異常ケースなどへの柔軟性がない。

(2) バッチ型

動作模擬内容をテストスクリプトとしてテスト項目毎に別途作成する。テスト項目の再利用が可能だが、作成にはテスト対象の動作を考慮

した分析が必要で、準備工数が多大となる。

(3) 対話型

動作模擬部が呼び出されるごとに、応答すべきデータ値をユーザに問い合わせる。ユーザ負担は少ないが、テスト実施時にユーザが介入するため、時間に依存するテスト項目に対応できない。また、再現テストが困難である。

長短所を表 1 にまとめる。

表 1 動作模擬方式の分類と長短所

	長所	短所
完全模擬型	テスト実施時の負担がない	テスト項目に対する柔軟性が不足
バッチ型	再利用が容易	準備工数(テスト内容の詳細解析)が多大
対話型	事前準備が不要	時間依存性がある 再現テストが困難

3.2 インタフェースの階層化

制御ソフトから見た場合、制御対象とのインタフェースは次のように階層化できる(図 2)。

(1) 通信インタフェース層

制御装置と中継装置との間の論理的通信路を管理する。

(2) 制御プロトコル層

制御対象機器へのアクセス権管理など、独自の制御プロトコルを管理する。

(3) 制御アプリケーション層

装置に対する制御指示の発行と、制御結果に応じた処理を実現する。

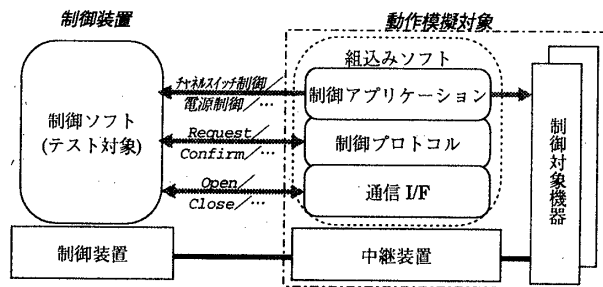


図 2 インタフェースの階層

3.3 機器動作模擬方式の検討

機器シミュレータは、制御ソフト全体のデバッグ・テストを実施するものと位置付けている。このため、通信インタフェース層・制御プロトコル層に関する詳細テストは、別手段にてテストする。したがって、機器シミュレータでは正常動作のみを完全に模擬し、当該層に関するユーザ指定を不要とした。

制御アプリケーション層に関して、多様なテストを実現するためにテスト項目を次の三項目に大別し、それぞれに適した方式を採用することとした(表 2)。

(1) 正常時動作の確認

動作模擬対象の正常動作を完全に模擬する“完全模擬型”とし、ユーザの介入を不要とする。

(2) 異常時動作の確認

異常時動作は多種多様で、全てを予め想定することはできないため、ユーザがプラグイン可能な“バッチ型”とする。正常/異常の区別は、模擬パラメタとしてテスト実施時に指定する。

(3) 模擬パラメタ変更に対する確認

正常動作/異常動作の指定、応答時間など、予め想定可能なテスト項目について、動作模擬内容の一部をパラメタ化し、テスト実施時に設定可能な“対話型”とする。

表 2 機器シミュレータでの動作模擬方式

階層	動作模擬方式	補足
制御アプリケーション層	完全模擬型	正常動作を模擬
	バッチ型	異常動作を追加可能 外部指定で切替え
	対話型	模擬パラメタを設定する UI を提供
制御プロトコル層 通信 I/F 層	完全模擬型 (正常時動作)	異常時動作など、詳細は別手段にて

4. まとめ

機器制御システム制御ソフト用機器シミュレータの開発構想を報告した。主な特徴をまとめる。

(1) 実システム構成方式

動作モジュールを模擬対象装置単位とし、実システム構成と同等のシミュレータ構成とした。

(2) Hybrid 模擬方式

テスト項目に応じて、完全模擬型/バッチ型/対話型からそれぞれに適した方式を採用した。

(3) プラグイン型異常動作模擬方式

異常時動作はプラグイン可能とし、ユーザの指定によりテスト実施時に呼び出す方式とした。

実現および適用については文献[1]を参照されたい。

文 献

[1] 渡辺 他：“機器制御ソフトウェア向けテスト環境の一事例 -実現と適用-”，第 58 回情報処大全講演論文集，講演番号 5C-08(1999, 予定).