

4K-5

組込みソフトウェア向け
ビジュアル試験システムの開発

木龍清治 深尾 至 西山好雄 豊島康文

富士通株式会社

1. まえがき

高度な情報サービスを提供するため通信分野における機器組込みソフトウェアは近年益々大規模かつ複雑化している。これらのソフトウェアを効率良く開発するため開発言語の高級言語化・視覚化が急速に進みつつあるがこれに合わせてシステム試験(実機搭載試験)時のデバッグ環境を高度化することが重要な課題となっている。我々は今回この課題を解決するための組込みソフトウェア向けシステム試験支援システムを開発した。

本稿では、はじめに組込みソフトシステム試験時の問題点について触れ、本システム構築のアプローチについて述べる。最後に本システムの概要と成果について説明する。

2. システム試験時の問題点

組込みソフトウェアが搭載される装置は、小規模であるため実機上に開発環境を載せられない。したがって、実機とは別の開発装置においてソフトを製造した後、実機に搭載し走行試験を行うという開発方法が一般に採られる(クロス開発)。このような開発形態では、試験段階において以下の問題が生じる。

①プログラムの見え方の違い(マシン語ギャップ)
製造時(高級言語)と試験時(マシン語)とでプログラムの見え方に隔たりがあり故障位置の特定と修正(デバッグ)に大変不都合である(図1)。

②試験作業時の煩雑な操作
実機に実装されるデバッグツールの機能は、主にラインコマンド入力やスクロール画面表示といった単純な機能しかなく、デバッグに必要な情報を得るために煩わしい操作を繰り返す必要がある。

以上の問題を解決する一つの方法は、専用の試験装置(インサーキット・エミュレータ)を用いることであるが、高価でありその機能をフルに発揮させるためには専門知識を必要とするため不特定多数の作業者が従事する大規模なソフトウェア開発には向かない。また、プログラムの視覚化という点では、従来の試験装置ではまだ不十分である。したがって、新しいアプローチが必要である。

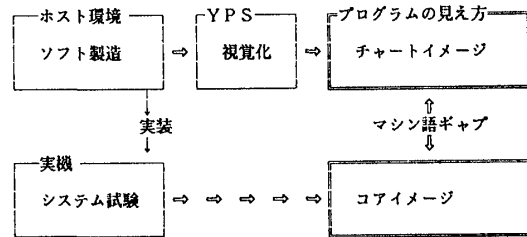


図1. マシン語ギャップ

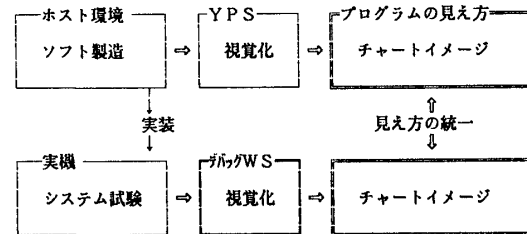


図2. 本システムのアプローチ

3. 新しい試験システム構築へのアプローチ

高解像度のディスプレイと高度なヒューマンインターフェースを備えた低価格のワークステーション(WS)が最近広く供給されるようになった。これらのWSをデバッグマシンに応用することは、システム試験への新しい実用的なアプローチに成り得る(ビジュアル試験)。我々は強力なデバッグWSを実機に配置することでシステム試験にともなう煩雑な処理を機械化しデバッグ作業を効率化することにした(図2)。

以下に本試験システムの狙いを記述する。

①ソフトウェアビューの統一

ソフト製造ホストでは、図形表現で視覚的かつ直観的なプログラミング言語(YPS *1)がサポートされている。デバッグWSにおいてもYPSをサポートすることにより、製造工程と試験工程でのプログラムの見え方を同じにする。

②ビジュアルデバッキング

視覚的な入出力操作機能をサポートすることよりシステム試験に伴う動的な実機の振る舞いを効果的に制御・監視できるようにする。

4. 試験システムの概要

本試験システムは、実機上で動作し被試験プログラムを制御するターゲットデバッガと、WS上で動作し操作表示機能を受け持つデバッグモニタと、製造ホストで動作し被試験システムの各種シンボル情報を自動的に収集するツールから構成される(図3)。本システムの主な特徴を次に示す。

- ① YPS指向のデバッグ制御と監視
YPSプログラムベースで実行中断点の設定・解除および日本語変数の参照表示が可能である。
- ② 表示情報の図形化と動画化
OS制御テーブル等が構造表示されるためデバッグに必要なデータを瞬時に理解することが可能である。また、CPU負荷率等を動画化したことによりシステムの動的な推移をリアルタイムに把握することが可能である。
- ③ マルチウインドによるデバッグ機能の同時利用
各々のデバッグ機能が一つ一つのウインドにより実現されている(図4)。それぞれのウインドは独立して同時に存在可能なので、必要なデバッグ機能を自由に組み合わせて使用することが出来る。
- ④ ポインティングデバイスによる操作の単純化
デバッグ作業に関わるほとんど全てのコマンド入力・選択がマウスにより操作可能である。

5. 試験システムの成果

本試験システムを実プロジェクトに適用した成果を以下に述べる。

- ① 製造工程と試験工程でのプログラムの見え方を一致させYPSベースでデバッグできるようにしたことにより、デバッグ時の思考の中断がなくなりプログラムの検証がより自然に行えるようになった。

② 表示情報を図形化し動画化したことにより、障害発生までの過程と発生の仕組みが一目で分かるようになりバグ解析の能率向上につながった。

③ マルチウインド機能を活用し関連するデータや情報を同一画面に表示したことにより、同じデータを何度も表示し直すというような無駄な操作がなくなった。

④ キーボード入力を極力減らし数値データの直接入力を除いてデバッグ作業のほとんどの操作をマウス選択で入力できるようにしたため、デバッグ操作が単純化され誰もがすぐに試験システムを使いこなせるようになった。

6. おわりに

組み込みソフトウェア向けのビジュアルな試験支援システムを開発した。システム試験にともなう煩雑な作業が機械化されたことと、製造工程と試験工程でのソフトウェアの見え方を一致させたことにより、作業者は仕様検証により多くの力を注ぐことが可能となった。また、実プロジェクトへの適用を通してシステム試験のビジュアル化はリアルタイムシステム開発の上で大いに効果があることが判明した。今後は、さらに上位の工程(仕様定義、設計)とのソフトウェア・ビュー(見え方)の統合を図るため本試験システムの改良を進めたい。

謝辞

本システムの開発に協力をいただいた多くの方々へ深く感謝します。

参考文献

- 1) 市川忠男, 平川正人: "ビジュアルプログラミング" bit, vol.20, No.4, pp.30, Apr. 1988.
- 2) Fujimoto, H. et.al.: "Visual Development Environment for Embedded Software", Proc., 36th ISMM, Jun. 1988.

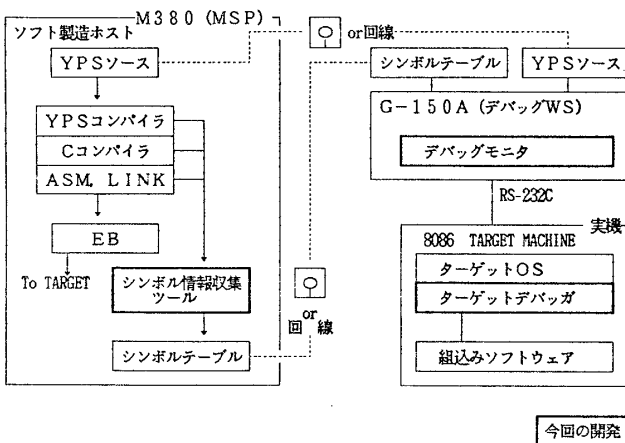


図3. 本試験システム全体構成

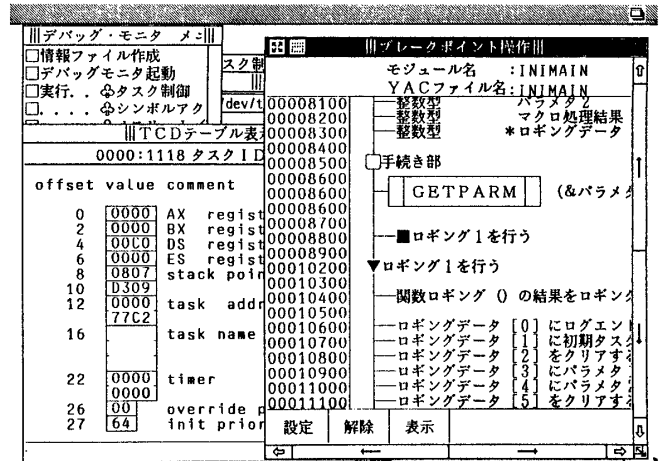


図4. 本試験システムの画面例