

状態遷移表導入による組込みS / W設計

静岡日本電気株式会社

パーソナルコミュニケーション 第1技術部

松井 大介

1. はじめに

ファクシミリの開発サイクルは、年々短くなってきています。これは、

- ・多機種少量生産化
- ・部品の寿命
- ・市場のニーズ

などが挙げられます。多機種少量生産は、技術の進歩が目まぐるしい現在、大量生産している間に次の機種が発売されて在庫の抱え込みを防ぐといった目的があり、避けられない状況となっています。また、メモリ等の電子デバイスは、大容量化が進んでおり、1年後、2年後にはすでに製造中止といったシーンも少なくありません。

これらの理由により、S / W開発は、短期間に高品質のものを仕上げなければなりません。開発期間が短いからと言って、適当な設計や評価をしていたのでは、市場に出てから致命的な障害が発生することは目に見えています。これらの要求を解決するために、状態 / イベントの追加が簡単にでき、シミュレーションやソースコード生成が容易にできることが、近道であると言えます。

2. ファクシミリ制御の特徴と現状

ファクシミリ制御の難しい点は、メカ動作、コードレスTEL制御、通信制御、ユーザオペレーションが複雑に絡んでいることです。ユーザオペレーション、通信制御、メカ制御が同時に行われるため、ちょっとしたS / Wの改造を行う際にも、全てを考慮した改造を施さなくてはなりません。例えば、ファクシミリ通信中の記録カバー開閉やコードレスTELからの呼出等が同時に起こるケースなどが挙げられます。しかしながら、以前の設計手法は、改造内容を羅列した概要設計からすぐにS / W製造に取りかかるといった非常に危険なものでした。当然、漏れや抜けが多発し、評価で発生する障害の大半は、それらの漏れや抜けが原因である障害でした。

3. 状態遷移表の導入

新規で設計する部分に関しては、状態遷移図と状態遷移表を使い分けることで、組込みS / Wの設計をしていました。状態遷移図の特徴は、状態の流れが分かりやすく、最終的にどの状態へ遷移するの

かが視覚的に分かります。その反面、設計や評価を行う段階になると、ある状態で予想をしない動作が抜けてしまうといった不具合が起こりやすくなります。状態遷移表では、状態の流れを視覚的に捉えることはできませんが、ある状態で発生するイベントが視覚的に分かるため、状態やイベントの漏れや抜けを確実になくすることができます。状態の流れは、マンマシン I / F 仕様書等の設計書でほぼ理解できるため、設計書を作成する場合には状態遷移表が最適と言えるでしょう。これらの理由から、新規設計部分に関しては状態遷移表を用い設計するようにしました。これにより、今まで設計書に現れなかった部分が確実に設計書に記述されるようになりました。また、設計レビュー時、レビュー、レビューの両方の立場から見てレビューが行いやすくなり、設計ミスレビュー段階で見つけることが容易となりました。ZIPCの導入は一部ではありますが、導入し、ソースコード生成まで行いました。

4 . 導入事例

今回はデータ通信の一部に ZIPC を導入しました。今回初めて使用したわけですが、状態遷移表の記述は EXCEL ライクで、楽に覚えることができました。

レビュー段階では、状態遷移表に詳しい処理まで記述されており、作成者の考え方やより細かい部分に対してレビューすることができました。ソースコード生成する段階では、ソースコードは人によって記述方法が異なるため、作成者以外が見ると、案外見にくいことがあります。しかし、ZIPC を導入することで、ある程度共通化されたソースコードが生成され、見やすいコードになりました。タスクの単体評価の段階では、作成した状態遷移表をそのままチェックリストとして用いることができ、チェックリスト作成の工数削減ができました。また、シミュレーションまでおこなったため、状態遷移に関するチェックする手間が省け、工数削減ができました。

5 . 今後の課題

現在の私どもの開発環境は、全てアセンブラにておこなっています。このため、問題点としては、生成されたコードをアセンブラにコンパイルし評価するわけですが、Cソースでの評価結果 = アセンブラに変換した結果となるかわからないことです。環境を整えることで可能となると思いますが、本開発環境ではこの点が問題であると考えています。