

# ZIPC +Perfect Pass+HILS による 自動試験環境適用事例

アイシン精機株式会社  
信頼性技術部 ソフトウェアセンター ツール開発チーム  
山田 信幸

## 1. はじめに

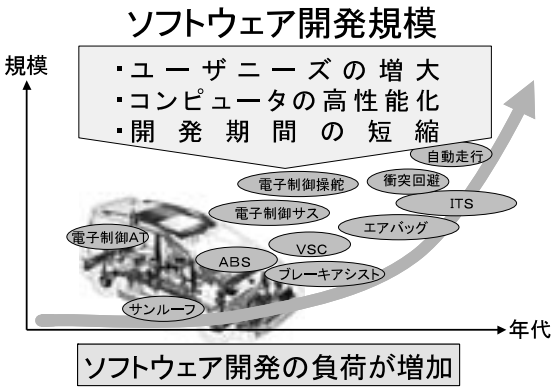
当社では自動車用システム製品の設計・開発・生産を行っており、各製品のコントローラーであるECU (Electronic Control Unit) に搭載するソフトウェア開発を行っている。

近年、チップの高機能化とユーザ要求の高度化により、図1に示す通りソフトウェアによる自動車の制御は高機能化し、プログラムの規模が加率的に増加し、複雑度も増してきている。

今後もITSなどの普及により、さらに高度な制御を組み込んだシステム開発が求められ、ソフトウェアの開発への負担は今後も増大することが予想される。

モデルベースでの試験自動化を積極的に進めている。

当社では様々なメーカーに様々な商品を提供しているため、社内全体を通してこのツール、この手法という統一はしていない。逆に考えると、ある手法を数々の製品にトライし、マッチするニーズに適用するといったことが可能である。ZIPC、Perfect Passについては以前より情報収集を進めていたが、今回社内ニーズにマッチする内容であり、これに合わせる形で試験環境の改善ができたので、事例として紹介する。



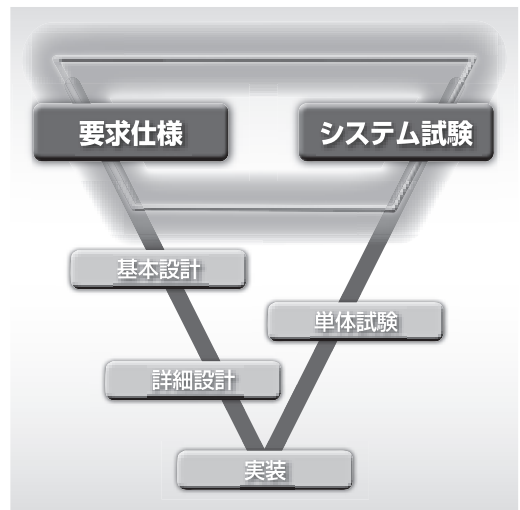
〈図1〉

## 2. モデルベース開発の現状

このような状況をうけて、当社ではソフトウェアのモデルベース開発を積極的に進めているが、車載用のECUはリソースの制約が大きい物も多く、オートコードを使用できない製品もある。しかし、このような製品でも仕様を記述したモデルの情報から試験の自動化は可能であり、

## 3. 適用領域

今回紹介する事例では、V字モデルの最上位部分に相当する〈図2参照〉。状態遷移図で記述された製品の動作仕様通りに実際のシステムが動作するかどうかを、Hardware In the Loop System (HILS) 上でシステム試験で確認する。



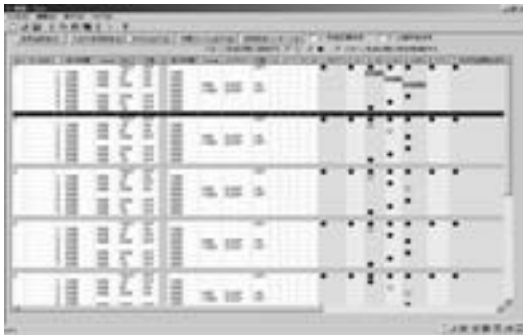
〈図2〉

## 4. 適用環境

今回紹介する事例では、富士通テン株式会社のHILSである「CRAMAS」を使用している<図3参照>。またCRAMAS用のツールとして、同社のSPOC（Signal Produce Of CRAMAS）という試験パターン自動作成ユーティリティを用いている<図4参照>。CRAMASで試験をするためのツールはいくつか存在するが、SPOCはON-OFFがはっきりしているデジタル信号の判定に強く、時系列データとして各種信号をECUに与え、戻ってきた値を自動的に判定させることが可能である。当社ではこのSPOCで作成した試験パターンをCRAMASで実行している。



〈図3〉 CRAMAS外観



〈図4〉 SPOC操作画面

## 5. 従来の試験方法問題点

従来は設計者が図形描画ツールで状態遷移図を作成していた。試験者はこの状態遷移図を元に、全ての状態から全てのイベントを抽出し、その結果どのような遷移、動作をするかを考えて試験条件表を作成していた。さらにこの1つ1つの試験についてSPOCの試験パターンを作成し、CRAMASにて実行していた。

この試験方法では、試験項目の作成と試験の実施に時間がかかっていた。その原因を分析すると、以下のものであった。

### ①記述のあいまいさ

状態遷移の考え方は一般的であり、UMLツールの導入前からUMLの状態遷移図に相当する仕様記述は行われてきた。しかし、モデリングツールではなく図形描画ツールを用いていたため制約が少なく、記述に関するルールに曖昧な部分が残っていた。その結果、試験条件の作成時に仕様確認／項目抽出のための検討時間がかかり、作業効率を落としていた。

### ②試験条件抽出工数

試験条件は試験者によるバラツキを防ぐため、全ての状態に対して全てのイベントを抽出している。ZIPCで例えると、全てのマスを試験する方法で実施してきた。この試験条件の抽出は状態遷移表からであれば容易にできるが、状態遷移図からではイベント、遷移に抜け漏れ等がないよう注意が必要となり、試験条件表の作成、レビューに膨大な工数がかかる。

### ③試験実行工数

試験条件を抽出した試験条件表の項目に対して、1試験ずつSPOCの試験パターンを作成している。SPOCは便利なツールであるが試験数が膨大であるため、これらの試験パターン作成にも相応の時間が必要となる。

従来の試験は、上記原因により試験に時間がかかっているが、製品の動作を確実に保証するため必要である。そこで、開発フェーズの途中でタイムリーな試験を行う場合に、効率よく試験を実施する方法がないか調査を進めていた。

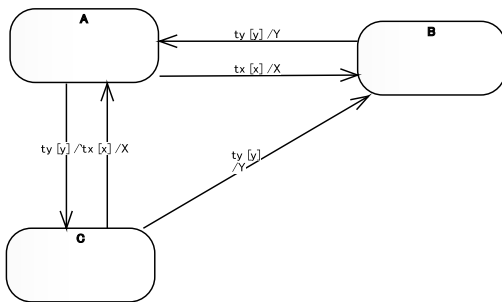
## 6. 自動化の検討

当社では、UMLモデルの記述にスパークシステムズ社のEnterprise Architect (EA) を多く用いている。ZIPCはEAとの連携が可能である事がわかったので、EAで記述した状態遷移図をZIPCに取り込み、ここからPerfect Passを用いることで、試験条件の抽出が大幅に効率化できないかと考えた。

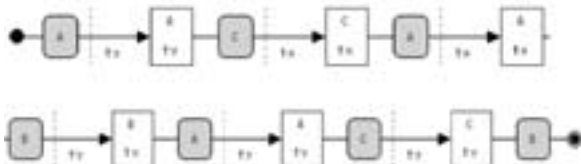
## 7. Perfect Passの機能

Perfect Passは状態遷移図の全アクションを実行する試験経路を、シーケンス行列検索／グラフ行列検索にて自動的に抽出することが可能である。状態遷移図のモデリングツールは、多くのツールベンダから出ているが、このようなPerfect Passに相当する機能を持ったツールは少ない。

例えば、図5に示す遷移をする状態遷移図の場合、全ての遷移を試験するA→C→A→B→A→C→Bといった試験経路が自動で抽出される<図6参照>。また、全ての遷移を試験するのに複数の試験が必要であれば、複数の試験も抽出される。この例は単純化したものであるが、実際の製品の状態遷移図は複雑かつ大規模なものであり、Perfect Passによる試験経路自動抽出は有効である。



<図5>

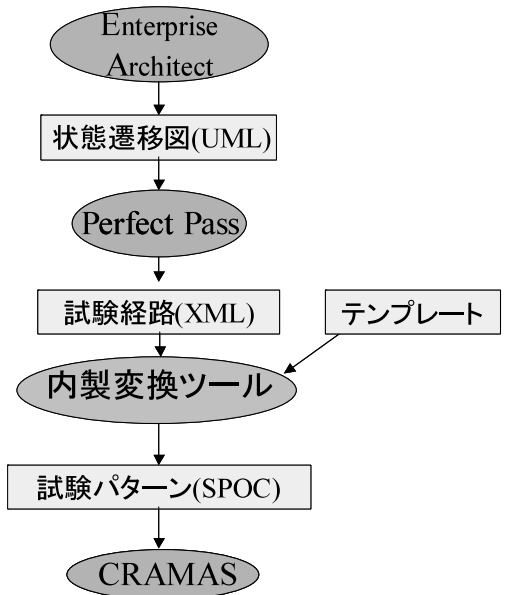


<図6>

## 8. 自動化の内容

作業フローを図7に示す。まず状態遷移図をEAで記述する。このモデルを入力として、Perfect Passで全ての状態を網羅する試験経路

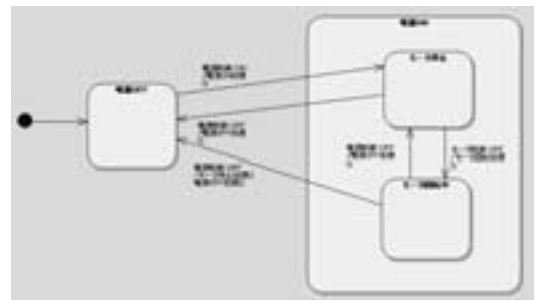
を作成する。Perfect Passは作成した試験経路をXMLファイルで出力する事が可能なため、最後にこのXMLファイルに対してテンプレートを当てはめることで、内製変換ツールがSPOCの試験パターンを生成する。図の作業フローは全体のつながりをわかりやすく説明するために、実際のツールの流れとは異なる。この点は後述する。



<図7>

実際の試験者の作業フローを画面イメージで説明する。

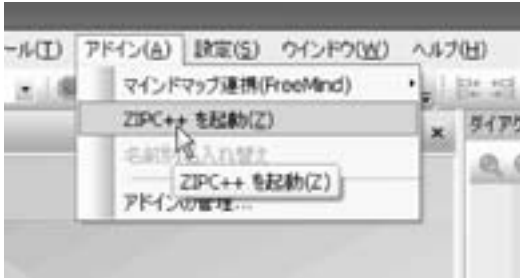
①EAで状態遷移図を記述する<図8参照>。



<図8>

②ZIPC++を起動する。

ZIPC++はEAのアドインとして起動できるので、メニューから呼び出す<図9参照>。



<図9>

③試験経路を作成する。

ZIPC++上で簡単なプロジェクト設定を実施後、Perfect Passで条件を設定し試験経路を作成する<図10参照>。



<図10>

④変換ツールで変換する。

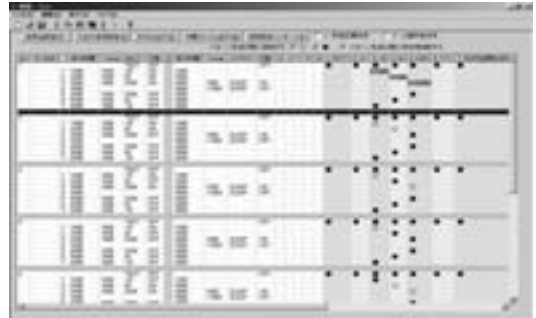
内製変換ツールにパスファイル名、テンプレートファイル名、出力ファイル名を設定し、SPOC用試験パターンに変換する<図11参照>。



<図11>

⑤タイミングを入力する。

自動出力されたSPOC用試験パターンに対して、各信号を発生するタイミングをSPOC上で調整する<図12参照>。



<図12>

⑥CARMAS上で、試験パターンの実行と判定を行う。

9. 課題と対策

9.1. 信号の入力タイミング

CRAMAS上で試験を実施するにあたり、試験環境上で信号を安定化するためのWAIT時間、信号間のタイミングの制約等、製品の動作仕様と直接に関係しない信号の入力タイミングの調整が必要となる。これらの情報を出力する試験パターンに載せるには、試験用情報を追加した「試験用モデル」の作成や、これら条件を考慮したテンプレートを当てはめる方法が考えられる。現状の試験環境では、様々な条件を考慮する必要があったため、テンプレートで定型的なタイミングを設定しておき、最終的なSPOCの試験パターンで調整することにした。

9.2. ツールチェーンの問題

試験の作業フローは8項で示した通りであるが、実際の作業はツールチェーンの制約で操作が多く作業効率が低かった。EAで作成したモデルをPerfect Passに取り込み、内容確認するまでの作業フローを図13に示す。

1. EAを起動しプロジェクトを開く。
2. EAにてZIPC++を起動させる。
3. ZIPC++にてEAのモデルを読み込む。
4. ZIPC++にて読み込むパッケージを指定する。
5. ZIPC++にてドキュメントツリーを確認する。
6. ZIPC++にてSCを登録しているパッケージを指定する。
7. ZIPC++にてコンバートメニューを選択する。
8. ZIPC++にてコンバートするSCをチェックする。
9. ZIPC++にて出力するSTMファイルを指定する。
10. ZIPC++にてSC⇒STMコンバートを開始する。
11. ZIPC++にてメッセージを確認する。
12. エクスプローラにてSTMファイルを確認する。
13. ZIPCを起動する。
14. ZIPCにてZIPCプロジェクトを新規作成する。
15. ZIPCにてタスクを作成する。
16. ZIPCにてSTMファイルを登録する。
17. ZIPCにてSTMの空白セルを無視にする。
18. ZIPCにてSTM種別をE型通常にする。
19. ZIPCにてSTMファイルを保存する。
20. ZIPCにてZIPCプロジェクトを保存する。
21. Perfect Passを起動する。
22. Perfect PassにてPerfect Passプロジェクトを生成する。
23. Perfect PassにてPerfect Passプロジェクトを確認する。

〈図13〉

このように多くの作業が必要であった。そこでキャッツ殿に問題を提起したところ、全面的な協力によりPerfect PassとZIPC++の機能を改

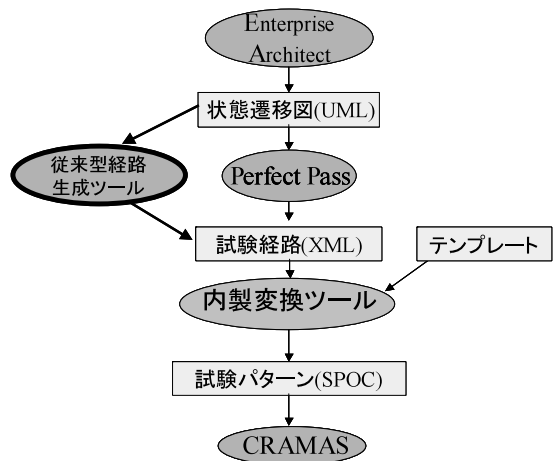
善していただいた。現在のバージョンでは、作業効率は大幅に改善し、図14に示す6ステップで完了できる。

1. EAを起動しプロジェクトを開く。
2. EAからZIPC++を起動させる。
3. ZIPC++にてEAのモデルを読み込む。
4. ZIPC++にて読み込むパッケージを指定する。
5. ZIPC++にてPPコンバートメニューを選択する。
6. Perfect PassにてPerfect Passプロジェクトを確認する。

〈図14〉

### 9.3. 試験抽出方法の問題

Perfect Passは全アクションを実行する試験経路を生成する。しかし、従来の試験ではアクションのない試験経路の試験も実施していたため、Perfect Passの試験経路と従来型の試験経路の両方の試験を試験できる試験環境が必要であった。そこで、従来型経路を生成するツールを内製で作成した〈図15参照〉。このツールで生成した試験経路を、Perfect Passと同じフォーマットであるXMLファイル〈図16参照〉で生成することにより、内製変換ツールを共通で使用できるようにした。



〈図15〉

```

- <pathGroup>
  <title>状態 - 2009/09/11</title>
- <path>
  - <state>
    <stateName>A</stateName>
  </state>
  - <transition>
    <eventName>tx</eventName>
    <op>x:X:B</op>
  - <state>
    <stateName>B</stateName>
  </state>
</transition>
- <transition>
  <eventName>ty</eventName>
  <op>y:Y:A</op>
- <state>
  <stateName>A</stateName>
</state>
</transition>

```

〈図16〉

## 10. 結果

ZIPC+Perfect Pass+ HILSを用いて、EAで記述した状態遷移図の自動試験ができるようになった。状態数が少ない製品ではあるが、実際に試験で活用し効果を上げている。

状態遷移図の記述はEAだけではなく、ZIPC、Stateflow等様々なモデリングツールを用途に応じて使用しているため、現在どのモデリングツールからも対応できるように、内製変換ツールの拡張を進めている。

## 11. 最後に

製品の複雑化、試験の複雑化が進む中、試験の現場ではPerfect Passを用いた試験経路自動抽出の必要性が増加している。積極的に活用して効率化を進めていきたい。

また、キャッツ殿には、当社の要望に対して真摯に対応いただけたことに感謝するとともに、よりよい開発環境の構築を共に進めていきたい。

## WATCHERS Back Numberで振り返るユーザ会



第11回のユーザーズカンファレンスでは、基調講演に名古屋大学 大学院情報科学研究科の高田広章教授より、組み込みシステム開発技術の方向性や自動車制御システムの課題、TOPPERSプロジェクトの取り組みなど、技術動向をご講演頂きました。また、カンファレンス開催前にプレスリリースされました、セイコーエプソン株式会社様のプリンタ開発へのZIPC適用事例も非常に注目を浴びました。

その他、幅広い分野で活躍する開発支援ツールの適用効果が発表され、ご来場頂いた方々には、終始熱心にご聴講頂きました。