モデルベース開発環境構築へ向けての NECエレクトロニクスの取り組みと キャッツ社への期待

NECエレクトロニクス株式会社
第二開発事業本部 第四システムLSI事業部
開発ツールグループ 開発マネージャ

橋本 一也

1．これからはモデルベース開発

近年、マイコン組み込み機器のソフトウェア規模や複雑さが増大傾向にあり、また商品化サイクルが年々早くなってきています。また、さらに最近では、企業のグローバル化が進み、設計・製造工程が海外へシフトしていき、海外拠点とのコミュニケーションや国内との作業の棲み分けに頭を悩ましているのが現状のようです。

このような問題に対し、解決を期待され、注目を向けられているのがモデルベース開発です。

モデルベース開発は、従来からあった考えですが、制御理論の進展、モデリング手法やシミュレーション手法の高度化またそれを支援する設計ツールの整備、といったことにより、より実用的なものになってきました。

特に、マイコン組み込みシステムでは、制御が中心であり、モデルベース開発は適した開発手法と思われます。

まずは、マイコン組み込みシステムにおけるモデルベース開発の概観を眺め、そのメリット、デメリット、モデルベース開発用ツールについて考えてみます。

尚、本稿で述べる「制御」、「モデルベース開発」、「HILS/SILS」などの定義や応用事例は、世間ではいろいろな定義があり、また実装の方法により位置付け・意味付けが異なってくることもあることを予めお断りしておきます。

2．制御システムとマイコン組込システムの制御

マイコン組み込みシステムにおける「制御」には、大きく「フィードバック制御」と「シーケンス制御」があります。

フィードバック制御とは、監視している制御量と目標値（基準量）との差を計算し、その差を小さくなるよう操作量を

変動させることです。

図 1 はフィードバック制御の典型的な構造を示しています。

制御器は、基準量と制御量との差を計算し、その差が小さくなるよう制御対象を操作します。この制御器にマイコンが使われることが多いようです。制御対象の制御量により制御系外（ここでは外界と呼ぶ）にも間接的に影響を与え、それがフィードバックデータとなることもあります。

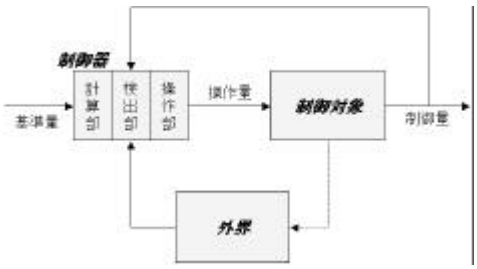

図 1 フィードバック制御の典型的な構造

一例として、エアコンを考えてみます。制御目的は、室温を設定した温度に保つことです。制御対象はファンやコンプレッサーです。

マイコンでファンやコンプレッサーのモータの操作量を制御することにより風量を調節します。室内温や外気温などの外界の状況に応じて、この風量を制御し、部屋の温度を上げたり、下げたりします。

また、省エネを実現するためは、こまめに ON/OFF をしなければなりませんが、頻繁に ON/OFF を繰り返すとモータが過熱したり騒音の源となるので、モータの回転数をフィードバックしながら、滑らかな始動、停止ができるように制御します。

このように、目標値（ここでは、設定温度や、滑らかに始動・停止できるモータ回転数の理論値）だけ与えれば、その目標値に近づくように制御して行くことになり、制御対象についてほとんど知らなくても、自動的に制御できるのがフィードバック制御の特徴です。

一方、「シーケンス制御」は、一連の決められた流れを自動的に行うものであり、マイコンが重要な役割を担っています。メカのコントロール、UI 部分や通信プロトコルなどがその例です。ロジック中心の設計になりますが、ロジックのヌケ・モレをチェックできる ZIPC などのツールが有効です。しかし、ロジック中心とはいえ、「シーケンス制御」においても、マイコンシステムに組み込まれたものを精度良く検証するためには、システムの周り、即ち制御対象や外界からのフィードバックが必要になる場合があります。

3．モデルベース開発と課題

● モデルベース開発

モデルベース開発は、まず制御しようとしている対象の動作を記述するモデルを作成し、そのモデルに基づいて、制御系を設計する手法です。

図 2 にモデルベースの開発プロセスを示します。

手順は、まず、ドメイン知識、実験データを基に、工学的問題モデルを駆使し、モデルを作成します。モデルは、制御器、制御対象、外界すべてについて作成します。作成したあとは、モデル上で確認を行い、モデル・パラメータ値の理論値を求めます。

このモデルをベースに制御器上の組込ソフトウェアを開発します。モデルからプログラムを生成するツールもあります。求めた理論値に近づくようにパラメータをチューニングして行きます。

また、並行して制御器のハードウェアを開発します。それぞれの工程で確認を行います。ハードウェアの確認を行う場合には、その制御対象および外界はプロトタイプとして作成します。

確認が終わったら実装し、実測値によりパラメータのチューニングを行います。

モデルベース開発では、各工程でそれぞれのレベルでの検証を行い、より上位レベルで検証することにより、下位工程での不具合による後戻りを少なくすることができます。つまり開発 TAT の短縮や品質向上につながります。

また、モデルベース開発では、デバッグが容易であることもメリットのひとつです。マイコン組込システムの実機では、制御対象に悪影響を及ぼすことから、デバッグ時などで実行停止させることがで

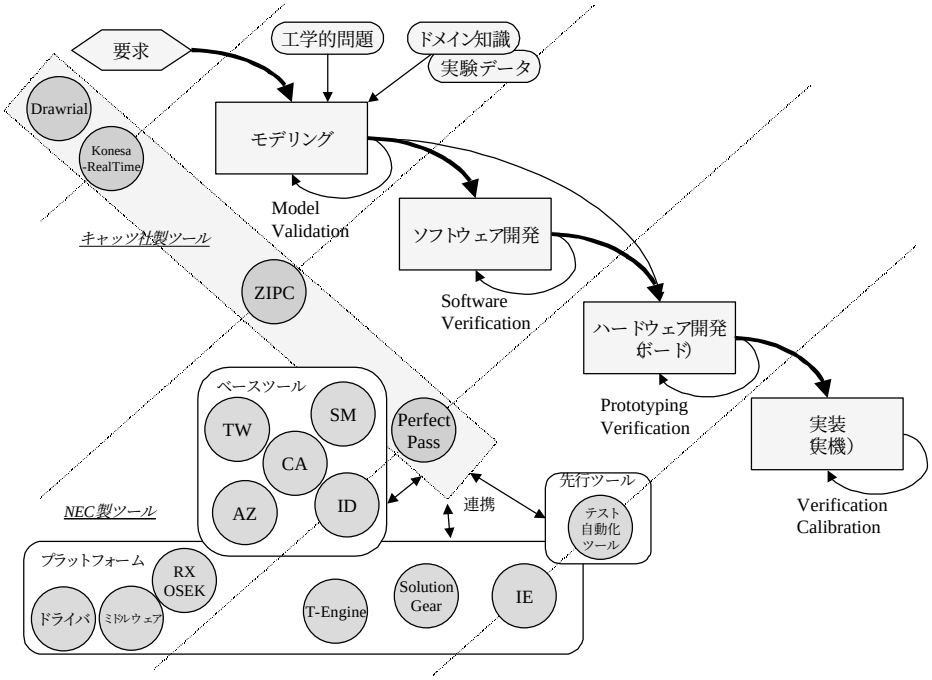

図 2　モデルベース開発プロセスとツール

きませんが、モデルならそれが可能です。任意のところでプログラムをブレークさせ、状態を確認するようなデバッグが可能です。他のメリットとして、モデル言語での記述は見た目にわかり易く、皆の理解が統一できるので、設計資産化の助けになります。そして、海外拠点とのコミュニケーションにも役立つことになります。

デメリットとしては、既存の資産が使えなくなる可能性があり、どうしても初期投資がある程度必要となるということです。メリットの効果を実績から定量的に出す必要性がありますが、デメリットは十分に補えるのではないかと考えています。

このように、モデルベース開発がマイコン組込システムに適していると言えるのですが、H∞制御に代表されるロバスト制御理論といった理論の進展により、少しくらい不正確でも制御性能がそんなに落ちないことが実証されたので安心してモデルが使えるようになったとはいえ、制御対象や外界をモデル化するのは簡単ではありません。少しくらい不正確でもよいと言っても、その対象の様々な特性を知っていなければならないと作れないからです。モデルのパラメータは数百～数万のオーダーになるといわれており、それらを全て正確に把握することは難しいのです。この大量のパラメータをどう決めていくかがポイントです。モデルの複雑化/高性能化に伴い、実際の動きを見て決めていく試行錯誤的なやり方では対応できず、理論的にモデルを求めていく方法に変わっていきました。

このように、モデルベース開発では、制御対象や外界をどう作成するかが大きな課題になります。

● HILS と SILS

モデルベース開発に関連して HILS（Hardware-In-the-Loop Simulation）や SILS（Software-In-the-Loop Simulation）という言葉が使われていますが、これはモデル検証の形態のことです。ここでは次のように定義します。

SILS は、制御器をソフトウェアで実装し（マイコンモデルでプログラムをソフトウェアシミュレーションする）、制御対象や外界はソフトウェアのモデルで検証を行う形態です（**図 3**）。

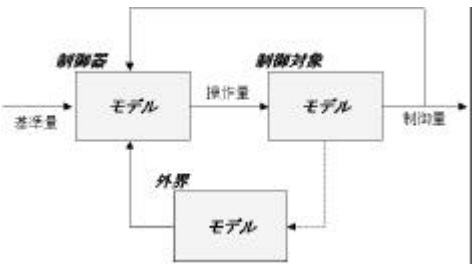

図 3　SILS の定義

HILS は、制御器をハードウェア（CPU ボード + プログラムなどで実装）で実現し、制御対象や外界はプロトタイピングのモデルで検証を行う形態で

す（**図 4**）。

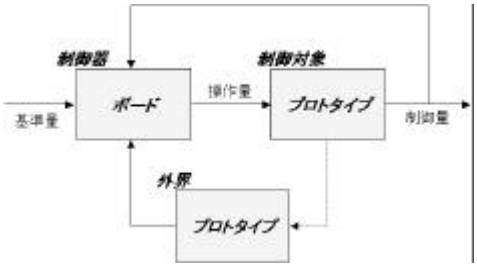

図 4　HILS の定義

HILS、SILS とも制御器のプログラム（組込ソフトウェア）は、モデルを参照して作成するか、モデルから自動生成します。モデルからプログラムを自動生成する際の課題をあげておきます。

① 性能情報の正確性の向上
デバイス、ライブラリ、またコンパイラにより、性能品質（速度、コードサイズ）が異なっても許容範囲に収めることが必要です。そのためには各種ソフト／ハード IP の性能プロファイルを入力でき、検証精度を上げるような工夫が必要です。

② デバッグ環境の構築
ツールが自動的に生成したコードはユーザには理解できない場合があり、できる限りモデル記述レベルでデバッグできるように配慮すべきです。そのためには、モデル記述ツールと下流ツールとの連携が必要です。

4．モデルベース開発への対応

モデルベース開発環境構築へ向けて当社の対応を説明します。

まず、NEC エレクトロニクス製下流ツールについて説明します。

NEC エレクトロニクスでは、V850 シリーズマイコンを中心に、78K シリーズマイコン、VR シリーズマイコン用のソフト開発環境をツールベンダー様の協力の基に整備しています。

NEC エレクトロニクスのソフト開発環境は「プラットフォーム」「ベースツール」「先行ツール」の大きく 3 つに分類できます。これらのツール群は相互に連携し、各ツールが持つ機能を効率良く使うことができるように配慮しています。

当社が提案するソフト開発環境は、

- 各ツールによりデバイスの性能を最大限に引き出します
- シームレスな環境により開発 TAT が短縮できます
- 高品質ソフトウェア部品の利用により品質を向上できます
- 拡張性があり、柔軟な対応ができます

といった特徴があり、お客様が安心して使える環境を目指しています。

● プラットフォーム

「プラットフォーム」では、OS、ミドルウェア、デバイスドライバ、評価用ボ

ードを中心としたソリューションを提供します。お客様セットを素早く製造して評価/機能確認していただくためのものです。OS は μITRON4.0（RX）や OSEK を中心に各 V850/78K/VR シリーズ用として品揃えを行っています。高い信頼性をもつ NEC エレクトロニクス製 OS や音声認識、TCP/IP、CAN/LIN ドライバなど標準的な機能を用意し、お客様セットを短期間で開発できるように支援します。

ボードソリューションでは、目的に応じて 3 種類のものを用意しています。「SolutionGear」は応用ソフトの開発を支援するための開発ツールキットです。ボード上に μITRON や各種デバイスドライバ、ミドルウェアを搭載し、すぐに使えるレファレンス環境として提供しています。また、CPU 選定時の基礎評価に使用できる「スタータ・キット」も提供しています。そして、ユビキタス時代の標準プラットフォームである「T-Engine」にも積極的に取り組んでいます。VR4131 用 μT-Engine と VR5500 用 T-Engine は、パーソナルメディア社から製品化済みです。

● ベースツール

「ベースツール」は、マイコンの性能を最大限に引き出すコンパイラ（CA）や洗練された GUI を持つデバッガ（ID）、また NEC エレクトロニクス製マイコンの高性能化に対応したエミュレータ（IE）を中心にしたツール群です。

さらに、周辺機能シミュレーションを充実したマイコンシミュレータ（SM）、業界で最初に製品化したパフォーマンスアナライザ（AZ）、またそれを進化させ、最適なメモリ配置を自動的に決定する性能チューニングツール（TW）といった特徴あるツールも用意しています。

現在、ベースツールについては、低価格化を進めており、お客様に導入して頂き易い製品体系を構築しています。

● 先行ツール

「先行ツール」は、現状のツール群にはない先端機能や、新しいコンセプトのツールです。その中のひとつですが、現在、検査の自動化を行なうことによりお客様の検査工数を大幅に短縮できる新ツールを具体化しています。

これからも新しいアイデアを盛り込んだツール開発にチャレンジして行きます。

●中流・上流ツールへの対応

図 2 は、モデルベース開発のプロセスとそれに対応したツールの位置付けを示しています。

中流・上流ツールはツールベンダー様のものを利用し、NEC エレクトロニクス製下流ツールと連動させ、お客様製品開発の上流から下流まで開発工程を支援します。

いくつか例を示します。ZIPC を使っ

たモデルベース開発環境の例です。

前述のフィードバック制御では、制御器と制御対象や外界が理想的に密接につながっていると想定しているのですが、実際は、遠く離れており、通信を行い操作量の伝達や制御量のフィードバックをかけることもあります。

このような通信路をもった機器の制御では、その通信路の遅延など考慮しなければなりませんが、比較的楽に検証環境を構築できるようになります。つまり、通信路を介することにより、連続系から離散系モデルとしてみることができるようになるからです。

応用例として車輌ネットワークを考えます。 **図5、図6** は ZIPC を使った HILS 環境です。

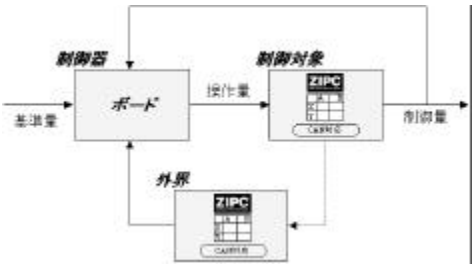

図5 ZIPC を使用した HILS 環境(例)

ZIPC は CAN に対応しており、パソコン上の ZIPC から NEC エレクトロニクス製 CAN カードを使って直接 CAN 通信路にアクセスすることができ、

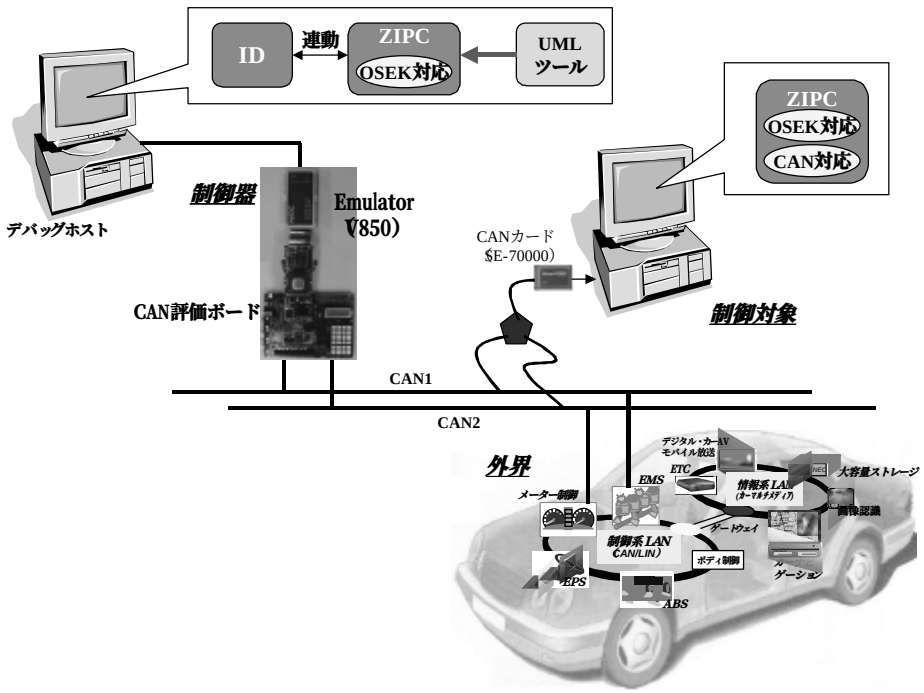

図6 ZIPC を使用した HILS 環境（イメージ）

制御対象や外界モデルを ZIPC で構築することができます。

また、当社では、IC カードシステムの開発環境としてモデルベース開発環境を提案してきました（**図7**）。

各フェーズでビヘイビア検証、ビヘイビアシミュレーション、シミュレーション検証といったモデルを使った記述・検証を提案しています。できるだけ前段階で検証を行い、検証精度を上げて行くやり方です。

この例にありますように、モデルベース開発において、ZIPC は効果的に使えると考えます。今後、適用効果を定量的に出して行きたいと考えています。

5．今後の取り組みとキャッツ社への期待

以上、モデルベース開発の課題とモデルベース開発環境の構築に向けての当社の取り組みを述べてきました。

最後に、キャッツ社製ツールへの期待を述べておきたいと思います。

キャッツ社の製品は、画面遷移図をベースにした Drawrial や Kosesa-RealTime など魅力的なツールが多いのですが、ここでは、最近注目されている PerfectPass に対して期待を述べておくことにします。PerfectPass は ZIPC の状態遷移表からとり得る全ての経路（テストケース）を抽出するツールです。モデルベースレベルで検証精

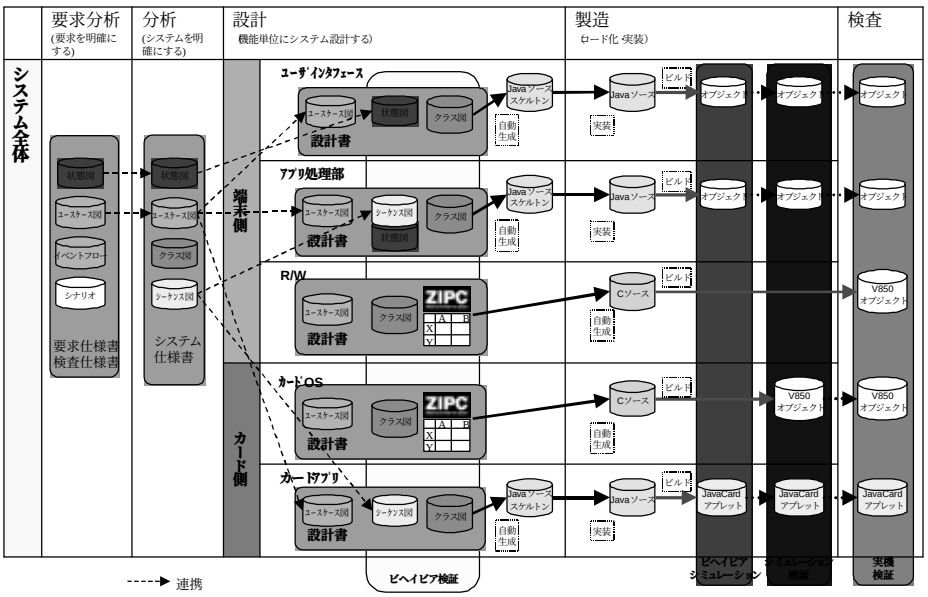

図 7　IC カードシステム開発プロセス

度を上げることにより、検査工程での手戻りを少なくすることが期待できます。当社では NEC エレクトロニクス製ツール群との連携を検討しているところです。

そこで、現段階の実装レベルに対して次の課題解決を期待したいと思っています。

ひとつは、PerfectPass で抽出した経路を人間の手を煩わせず自動で検査する検査エンジンとの連携です。小さいマトリクスでも数千万項目におよぶ膨大なテストケースの確認は時間がかかることが予想されます。この大量の経路を如何に効率よく最適化し、テストケースをリダクションできるかがポイントになるのではないかと思います。

もうひとつは、問題個所を特定する機能の実装です。

「経路抽出→検査→問題個所の抽出→修正→‥」

といった利用サイクルが最小時間でできるような操作性を実現して欲しいと考えます。

また、PerfectPass だけではなく、先に述べた設計・製造工程の海外シフトの対応として、キャッツ社製ツールの海外サポートの強化をお願いしておきます。

今後とも、NEC エレクトロニクスでは、キャッツ社をはじめ強力なツールを持つツールベンダー殿と連携し、最先端の環境をいち早く提供して、お客様の価値創造に役立てていただけるよう努力してまいります。

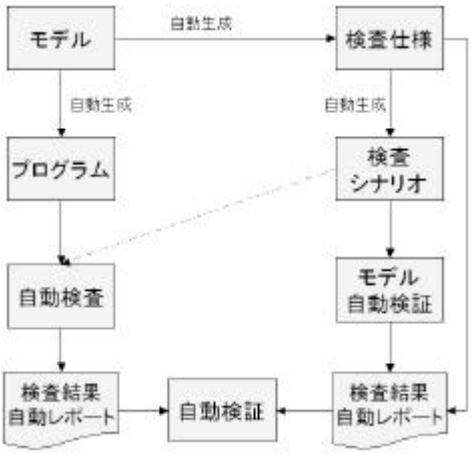

図 8　モデルベース開発プロセス(目標)

図 8 はモデル開発から検証までの作業フローであり、近い将来の目標です。モデルを作成して検証できたら、あとは自動でソフトウェアの検証まで行うというものです。要素技術は確立されつつあるので、近々に現実となる可能性は高いと感じています。当社もツールベンダー殿と連携し、実現に向けて努力して行きたいと考えています。