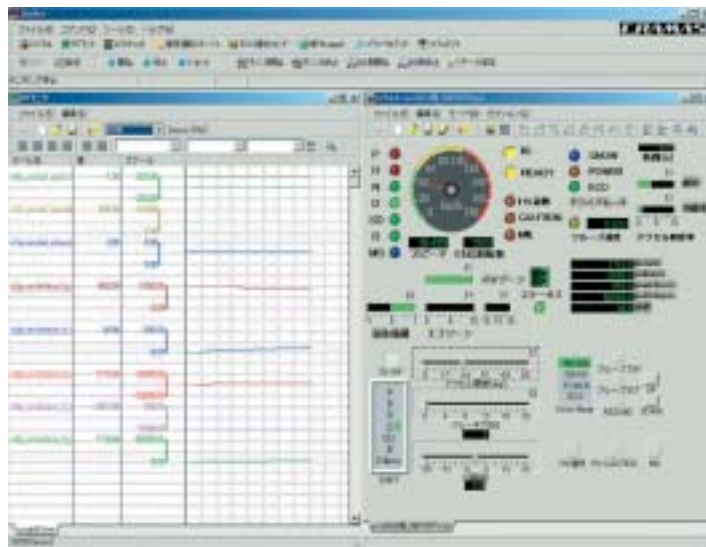


制御アプリケーション仕様開発へのSILSの適用

Use of SILS to Development of Control Application Specifications

細	川	健	司	Kenji	HOSOKAWA
吉	田	光	男	Mitsuo	YOSHIDA
松	尾	智	裕	Tomohiro	MATSUO
鷹	取	剛		Tsuyoshi	TAKATORI
高	橋	淳	二	Junji	TAKAHASHI



要 旨

従来、制御アプリケーションの仕様検証場面ではHILS（Hardware In the Loop Simulator）を用いていたが、仕様開発局面ではECU（Electronic Control Unit）が存在しないことが多い。その場合、類似システムのECUを改造して使用していたが、検証準備工数を押し上げる要因となっていた。

このような状況のなか車両メーカー間の開発競争は厳しさを増しており、市場の変化に迅速に対応するため、さらなる開発期間短縮が求められている。そこでHILSをさらに一歩進め、ECUハードウェアをソフトで置き換えた仮想ECUを使用するSILS（Software In the Loop Simulator）を導入することで、仕様開発検証工数の短縮を図る。

今回、制御アプリケーション動作に必要な信号をプラントモデルとRAM値で授受するISSレスシミュレーション方式のSILSを、エコインジケータアプリケーション仕様開発場面において実際に適用したので報告する。

Abstract

Conventionally, HILS (Hardware In the Loop Simulator) was used for verification of control application specifications. However, ECU (Electronic Control Unit) was rarely completed by the time of specification development. In that case, we modified a similar existing ECU and used it, but that increased the verification preparation man-hours.

Under these circumstances, due to the more intensive development competition among car manufacturers, we need to shorten more the development period of products to respond swiftly to market changes. To reduce the specification development man-hours, we upgraded the HILS to SILS (Software In the Loop Simulator) equipped with virtual ECUs that replace ECU hardware of HILS.

We used the SILS to the development of eco-driving-indicator application specifications. This paper describes the SILS employing the simulation method that does not require ISS and signals for the control application operation are sent to / received from the plant model in RAM values.

1

はじめに

車両に対する環境・安全への要求の高まりを受け、車両制御システムは年々複雑化・大規模化が進んでいる。また、車両メーカー間の開発競争は厳しさを増しており、市場の変化に迅速に対応するために開発期間の短縮が重要課題になっている。

この開発期間短縮に対応するため、制御アプリケーションの仕様開発場面で用いる検証ツールも進歩を遂げ、近年では実機を使用せず全てをソフトウェアシミュレーションで行うSILS (Software In the Loop Simulator) が各社で開発されている。

本稿では自社開発のSILSを、制御アプリケーションの仕様開発場面において実際に適用したので、その技術および検証結果について紹介する。

2

仕様開発へのSILS導入の経緯

2.1 仕様開発環境の変遷

一般に制御アプリケーションの仕様開発は①仕様設計、②ソフト作成、③検証というループを廻して完成度を上げていく(図1)。

以降、「検証」とは、仕様開発における検証の意味で使用する。

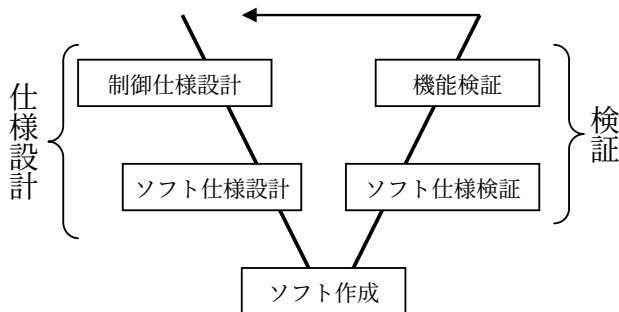


図1 仕様開発検証ループ

Fig.1 Verification Loop of Specification Development

ここで「③検証」ツールの変遷を見ていくと、シミュレーションツールが存在しない時代には実車を用いて検証を行っていた。しかしこれは実車の準備、評価用ECU (Electronic Control Unit) の製作などで一連の検証ループにかかる時間が大きい上、実車の破壊、暴走などの危険もあり、机上で充分に安全性を確認する必要があった(図2)。

その後、この問題を解決するために、実車相当のシミュレーションをPC上で行うHILS (Hardware In-the Loop Simulator) ツールが登場した。これにより大幅に検証にかかる工数は削減され、机上で効率の良い検証が可能になり、車両を破損する危険もなくなった。当社においての

HILSシステムはCRAMAS (Computer Aided Multi-Analysis System) の名称で自動車メーカ、サプライヤグループ向けに製造・販売を行っている。

HILSはECUを使いシミュレーションを行なうが、制御アプリケーションの仕様開発局面ではECUが存在しない事が多い。そのため、従来は類似システムのECUを改造して使用していたが、これは検証準備工数を押上げる要因となっていた。

そのような状況の中、更なる開発期間短縮が求められるようになり、ECUやECUとの信号の授受を行うベンチシステムなど、すべてをソフトウェアでシミュレーションするSILSが登場した。ソフトウェアのみで構成されるためHILSに比べ安価であり、1人1台のシステム利用環境が狙え、さらにECU、ハーネス、計測ボードなどのハードシステム準備にかかる工数を短縮することが可能となった。

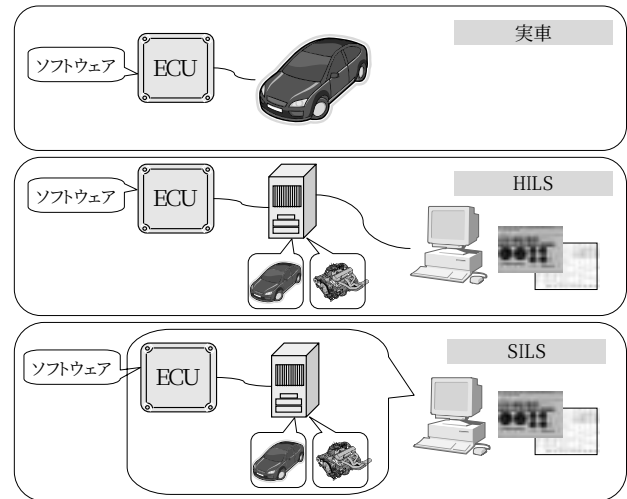


図2 検証ツールの変遷

Fig.2 Changes of Verification Tool

2.2 当社SILSの種類

当社においては、CRAMASの技術を応用し、用途別に以下の三つのバリエーションのSILSを開発した。ユーザーは必要とする速度や精度に応じてSILSタイプを選択して使用する。

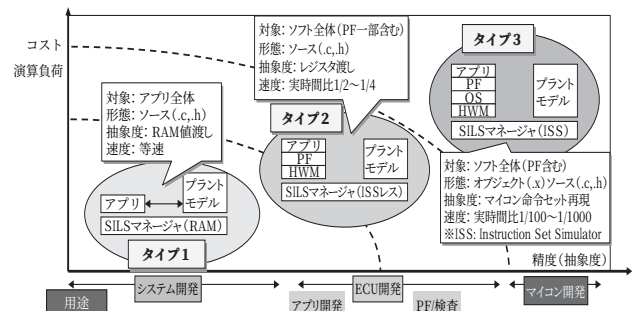


図3 当社SILSの種類と特徴

Fig.3 Types and Characteristics of Our SILS

このうち、仕様開発では、演算負荷が軽くアプリケーション単体の実行に向く、SILSタイプ1を使用する。

参考までに、タイプ2はECUソフトウェア全体のシミュレーションを実行する時に、タイプ3はマイコンシミュレーションを厳密に行ないたい時に使用する。

3 エコインジケータの概要

ここで、今回、仕様開発にSILSを適用した制御アプリケーション「エコインジケータ」の概要を説明する。

当アプリケーションは近年の地球温暖化対策であるCO₂排出量の削減・化石燃料の使用量削減のためにエコドライブを実施できるように支援する機能を有すもので、自動車メーカ各社が順次展開している機能である。

走行時の燃料は発進時・巡航時に多く消費されるため、この時の無駄な操作（アクセル急踏込みによる急加速・アクセル変動による速度変動）をドライバが抑止できるように支援する。

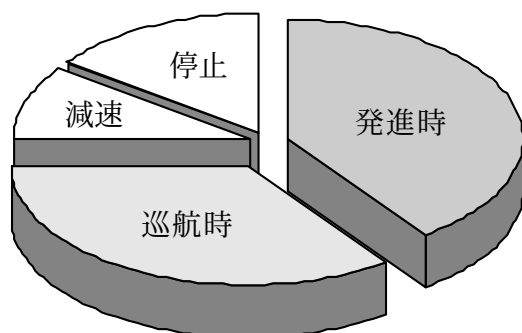


図4 都市部での走行モードと燃料消費割合
(データは(財)省エネルギーセンタHPより)

Fig.4 Driving Modes in a city and Fuel Consumption Ratio
(From the website of The Energy Conservation Center, Japan)

ドライバが速度域に応じたエコドライブをしているかを、現在、車速とアクセル踏込み量などから判断し、結果をランプなどの表示デバイスを通じてドライバに通知することでエコドライブを支援するものである。

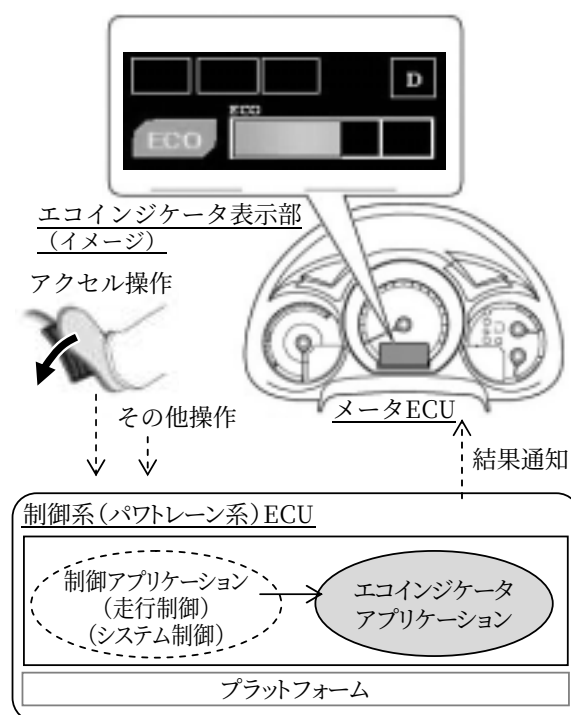


図5 エコインジケータ概要図
Fig.5 Outline Drawing of Eco-driving-Indicator

当社はこのアプリケーションの仕様開発をトヨタ自動車株式会社殿と共同で実施しており、その開発でSILSを活用した。

従来通りであればこの仕様開発も、類似システムのECUを改造しHILSを用いて開発するところだが、実機レスでの新たな開発手法への挑戦と以下①～③の理由から、SILSを適用することにした。

- ①開発期間短縮
- ②数多くの仕様を短時間で作成・検証
- ③多人数で同時に作成・検証するため安価な検証ツール

4 SILS概要

ここで仕様開発者を支援するために今回の開発で活用したSILSの概要を紹介する。

全体構成は図6ようになる。

ユーザアプリケーションは専用PF (Platform) と時間管理を行うイベントマネージャとが統合され、一つの実行体としてVC (Microsoft Visual C++) でビルド・実行される。

実行体からはVirtual CRAMAS DLL (Dynamic Link Library)の接続も行っており、DLLを通じて車両モデルなどを実現したプラントモデルや、CRAMAS GUI (Graphical User Interface) との入出力信号を授受している。

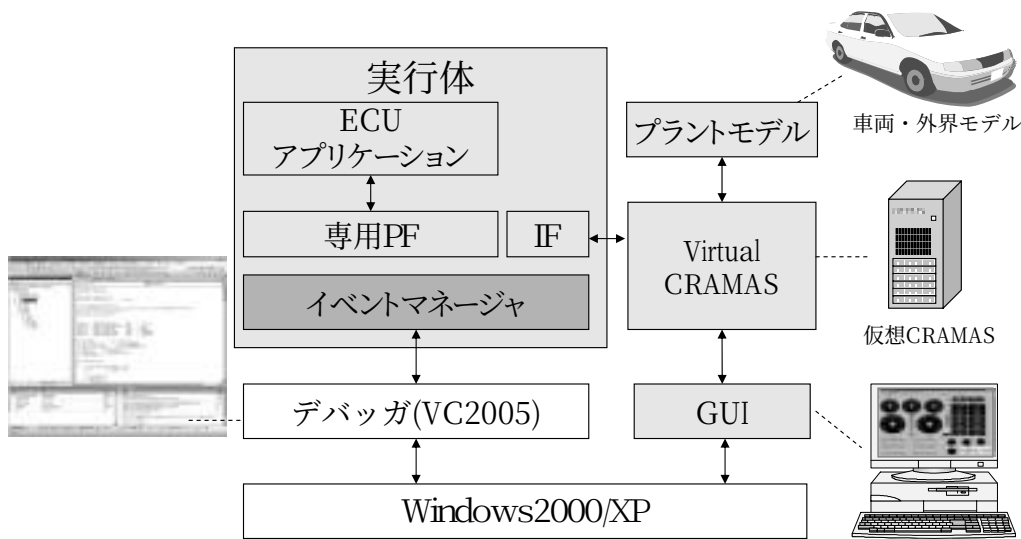


図6 SILSブロック図
Fig.6 SILS Block Diagram

つぎにそれぞれの構成部について説明していく。

①ECUアプリケーション

仕様開発を行うソフトウェアそのもの。

このアプリケーションには基本的に手を加えなくてよい構成となっている。

②専用PF・IF

ECUアプリケーションをSILS上で動かすためのインターフェース部。ECUアプリケーションを変更不要にするための仕組みを提供。

③イベントマネージャ

当社独自開発のシミュレーションマネージャ。

先行的なアプリケーション開発を行う上で、仕様不備など問題発生時にも迅速に対応できるよう、システム中にブラックボックスを作らない目的で開発された。

イベントマネージャはSILSシミュレーションの時間管理を司っており、この機能によりアプリケーションの定期/非定期処理を時間矛盾を起こすことなく、高速に処理することが可能となる (図7)。

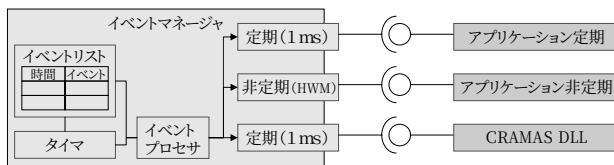


図7 イベントマネージャ構成図
Fig.7 Event Manager Configuration Diagram

イベントマネージャの内部にはSILSシミュレーションにおける絶対時間となるタイマブロックと、イベントとその発生時間をペアで管理するイベントリスト、それら进行处理するイベントプロセッサがある。このイベントプロセッサが定期/非定期処理から呼ばれる複雑な時間起動要求を管理

することで、割り込み処理のシミュレーションまで可能となった。

④デバッガ (VC2005)

非組み込み系のデバッグ環境では標準的に搭載される機能として、ブレークポイントによるプログラムの実行停止やステップ実行、RAM値を書換え後の再実行などがある。しかしHILSではハードウェアを使用しているため、プログラムを停止してもマイコン以外のハードウェアは動き続けており、再開させた時に停止させた状態を維持できないため、検証を継続することができない。

一方、SILSではブレークポイントを始めたVCの豊富なデバッグ機能が全て利用可能になる。さらに、プログラム停止後の再開に関しても、ハードウェアが同時に止まるため、あたかも全ての時間を止めていたかのように元のまま再開することが可能である。



図8 VCデバッガイメーシ図
Fig.8 VC Debugger Image

⑤プラントモデル

アプリケーションが制御対象とする車両などは、プラントモデルとして組付けられる。

⑥Virtual CRAMAS / GUI

シミュレーションの開始終了や、データの入出力を行う。実際の画面イメージを図9に示す。

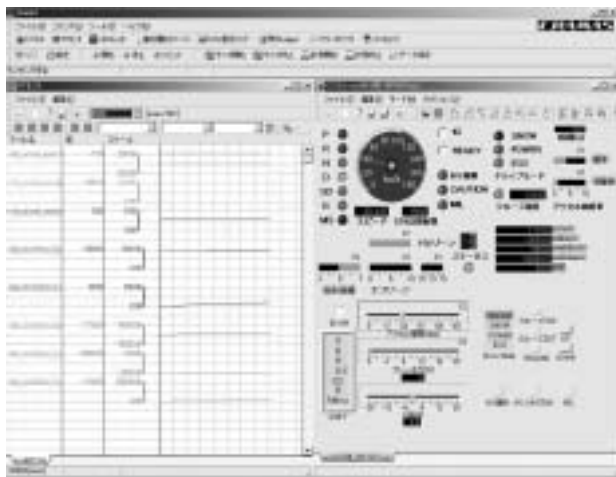


図9 実際の画面イメージ
Fig.9 Image of Actual Screen

5

SILS活用事例

今回適用したSILSと、エコインジケータ仕様開発での活用事例について紹介する。

5.1 エコインジケータ仕様開発への活用

今回適用したSILSタイプ1では制御対象となるプラントモデルとのデータ授受をRAM値で行う。そのため、従来のHILS用プラントモデルとアプリケーションの間にインターフェース層を設けている(図10)。

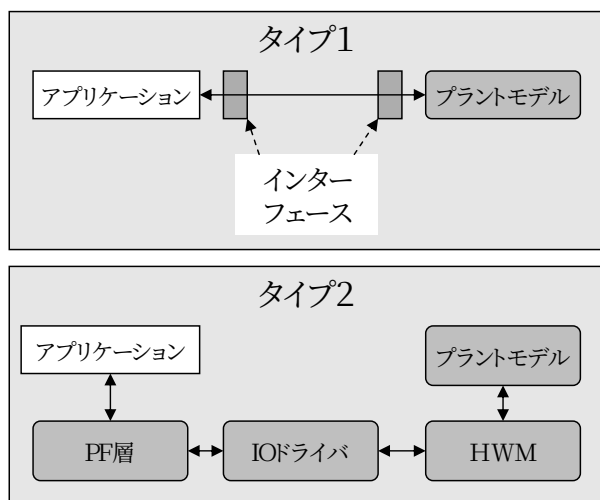


図10 タイプ1,2の相違点
Fig.10 Difference between Type 1 and Type 2

エコインジケータ仕様開発においてはターゲット車両が複数有り、それぞれでインターフェースが異なるが、それに対する汎用性の検証に好都合であった。例えば、プラントモデルからアプリケーションに入力として追加したい信号がある場合に、仕様開発担当者が、ハードウェアの知識が必要なHWM (HardWare Model) を作成することなく、インターフェースの変更だけで容易に追加することが可能である。さらに図11のようにそれぞれの担当が並行作業することで効率的に検証を行っている。

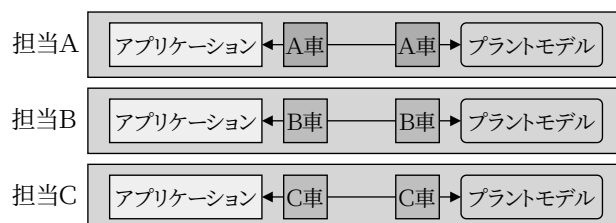


図11 タイプ1のインターフェース層の活用
Fig.11 Utilization of Interface Layer of Type 1

次項からは、エコインジケータ仕様開発に使用したSILSツールを紹介する。

5.2 GUI

メイン画面は大きく分けてつぎの3部分を使用する。

(1) 実行制御部

シミュレーションの開始・終了や計測の開始・終了などSILS動作全般を指示する。

(2) RT (RealTime) モニタ部

任意のRAM信号を時系列で表示する。

実時間(シミュレーション仮想時間)で刻々と変化する波形をモニタする。

(3) C (Control) パネル部

アプリケーションへの入出力用インターフェース。

ユーザがGUI部品を任意に組合わせて作成することが可能。

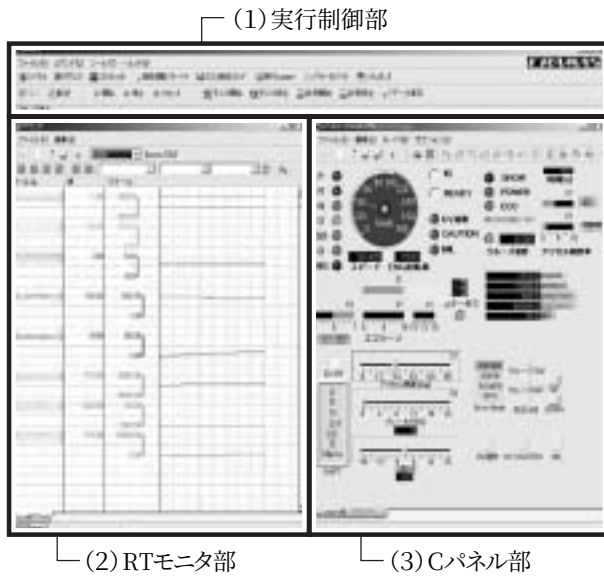


図12 SILS GUI画面
Fig.12 SILS GUI Screen

さらに、(3) Cパネルのエコインジケータ向け画面について説明する。

(A) 出力部

車速、エンジン回転数、各種状態表示ランプ、エコインジケータおよび仕様開発者がモニタしたいRAMの物理値などを表示している。エコインジケータは、実際のターゲット車両の表示部を複数作成し使い分けることで、より最終形に近い検証環境を実現している。

(B) 入力部

シフト入力、アクセル開度、ブレーキ力、各種モードスイッチなどを備えている。

上部の出力部がリアルタイムで変化していくため、これを見ながら下部の入力部で車両操作を実行する。

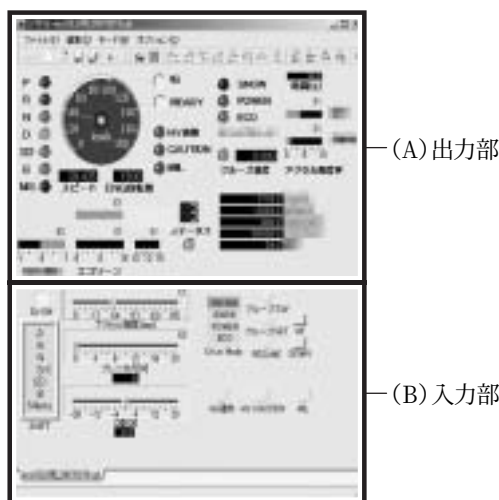


図13 GUI (入出力部)
Fig.13 GUI (Input / Output Segments)

5.3 プラントモデル

プラントモデルにはエコインジケータの制御に影響するつぎのブロックが含まれている。

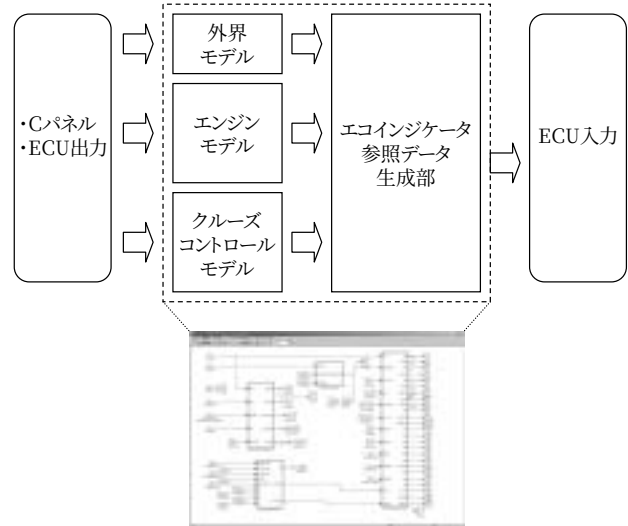


図14 プラントモデルブロック図
Fig.14 Block Diagram of Plant Model

・エンジンモデル

現在の車速、エンジン回転数を持ち、アクセル開度、ブレーキ力などから車両状態を変化させるモデル。

・外部アプリケーションモデル

クルーズコントロールなどエコインジケータとして影響を受ける他のアプリケーションモデル。

・外界モデル

勾配、路面摩擦係数など車両挙動に影響を与える外部環境を演算するモデル。

6

効果の確認

今回、ECUのない状態でSILSで仕様開発を行ない、最終的にはECUが完成した段階で実ECUへの組付け作業を行っている(図15)。その際、アプリケーションとしての機能に問題を起こさず、実ECUに合わせたインターフェースの調整だけで組付けを完了できている。

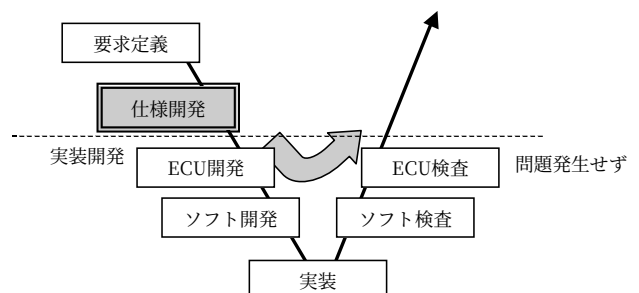


図15 仕様開発からECU検査までのV字工程
Fig.15 V-model Process from Specification Development to ECU Inspection

これは、ECUがない状況においてSILSが実機同等の機能を提供できた証であり、大きな成果だといえる。

特に、各仕様書同士の不整合や背反確認については、実際の動作を見て確認を実施しないと顕在化しないことが多く、この点について早期に課題を抽出、余裕を持ってお客様と調整、対策を検討できたことも大きな効果である。

また、安価な検証ツールという点では、VC2005 (ExpressEdition) はMicrosoftから無償提供され、その他の構成要素は自社製ということもあり、非常に安価に構成されている。これにより今回、仕様開発者1人に1台のSILSを提供できており、安価な検証ツールの提供という問題も解決している。

7

終わりに

今回対象としたエコインジケータシステムは他のパワートレーン制御に比べ、比較的小さいシステムであったため、SILSの構築を比較的容易に行うことができた。今後他の制御にて活用する場合、これら構築にかかる時間をいかに短時間で確実に実施できるかが活用の鍵であると思われるため、今回実施して顕在化したSILSについての課題や問題点を確実に解決しておく必要がある。

今回の挑戦により、実機のないアプリケーション仕様開発時にも、短期間でSILS開発環境を用意し、完成度の高いアプリケーション仕様開発技術を身に付けたといえる。

しかし今後も拡大し続ける制御の増大と、それに伴う開発期間短縮に対抗していくためには、今回のようなシミュレーションツールを積極的に取入れ、実践し、展開していくことが必須であり、このSILSを仕様開発における標準ツールとして完成させるよう継続して活動を行っていきたい。

筆者紹介



細川 健司
(ほそかわ けんじ)

1990年入社。以来、開発環境技術および設計支援ツールの開発に従事。現在、AE本部 先行開発統括部 先行システム開発部に在籍。



吉田 光男
(よしだ みつお)

1997年入社。以来、自動車用電子機器の開発および次世代のソフト開発プロセス構築に従事。現在、AE本部 制御システム開発統括部 次世代制御開発プロセス企画室に在籍。



松尾 智裕
(まつお ともひろ)

2001年入社。以来、自動車用電子機器の開発に従事。現在、AE本部 先行開発統括部 先行システム開発部に在籍。



鷹取 剛
(たかとり つよし)

1993年入社。以来、自動車用電子機器の開発に従事。現在、AE本部 先行開発統括部 先行システム開発部チームリーダー。



高橋 淳二
(たかはし じゅんじ)

1984年入社。以来、自動車用電子機器の開発に従事。現在、AE本部 先行開発統括部 先行システム開発部部長。