

2020年の組み込みソフトウェアとCASEツール

CATS社 取締役副社長

渡辺 政彦

はじめに

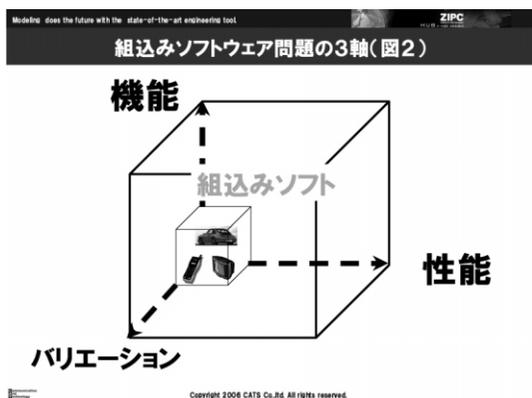
本稿では、2020年の組み込みソフトウェアがどうなるかを予測し、その開発を支援するCASEツールがどうあるべきかを述べます。

巨大・複雑化している組み込みソフトウェアは、QCD（品質、コスト、納期）が守れなくなってきました（図1）。



組み込みソフトウェアの問題

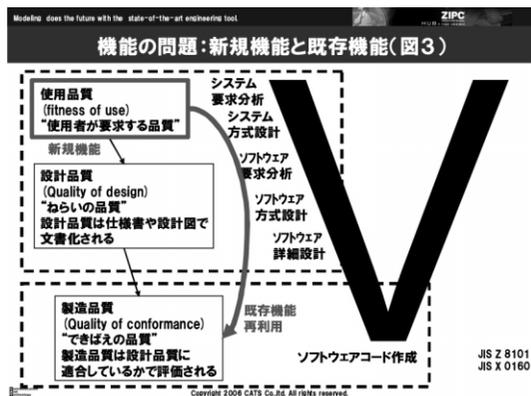
組み込みソフトウェアの巨大・複雑化の問題は、機能、バリエーション、そして性能の問題に分解できます（図2）。



機能の問題

機能における問題を図3に示します。機能の問題は、新規機能問題と既存機能問題に分けることができます。新規機能問題は、従来ない機能やサービスを実現するため、膨大な想定外の事象に対応する工数から納期問題や品質問題が起こることです。

既存資産の問題は、これまで何事もなく動いていた機能を、新機能と同居させたたん、何が起こるか予想もできないといった事態が起こることです。既存機能の開発者がいなくて、予想外の動作について対応が取れない状況にもなりがちです。

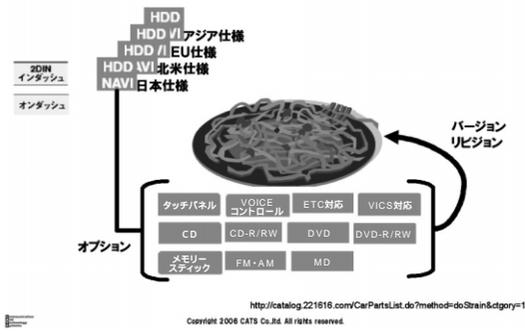


バリエーションの問題

バリエーションは、オプション、ワールドワイド展開、さらに対応する各種規約や不具合修正によるバージョン・リビジョンなど、多種多様化しています。組み込みソフトウェアのプログラム構造がきれいであっても、このバリエーションコントロールが整理されていないと、ソフトウェア構成がスパゲティになってしまいます（図4）。

バリエーションの多様化は、前述した機能の問題と相乗し、システム全体が理解不能になる危険性をもたらします。

Modeling does the future with the state-of-the-art engineering tool. ZIPC
バリエーションの問題:多様化(図4)

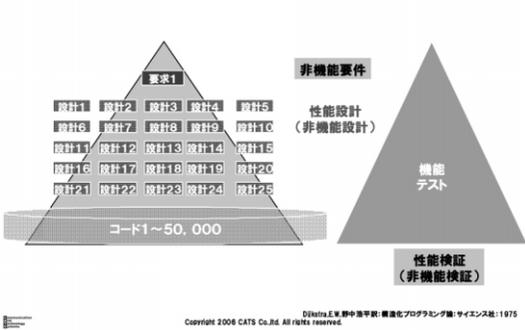


性能の問題

性能に関する問題を図5に示します。従来、機能分割など、機能に対する設計や検証(テスト)は、議論されてきました。ところが、組込みシステムでは重要な、性能設計や性能検証はどうでしょう。性能は、非機能としてあつかわれ、要件のみを、非機能要件として仕様書に記載するだけです。後は機能を実現するコードで、性能試験を行い、性能がでるまで、コードを修正します。

新規機能の性能設計・検証のほか、既存機能を修正した場合の、再度の性能評価や、バリエーションごとの性能評価も、膨大な工数を必要とします。

Modeling does the future with the state-of-the-art engineering tool. ZIPC
性能の問題:性能モデル設計と検証(図5)



組込みソフトウェアはどうなる

従来、組込みソフトウェアでは、機能の部品化、再利用を促進するためのソフトウェアプラットフォームはありませんでした。

2020年頃には、図6に示すように、組込みのプラットフォームとして、ハードウェアプラッ

トフォーム(以後HWPと呼ぶ)とソフトウェアプラットフォーム(以後SWPと呼ぶ)の2つのプラットフォームが存在することになるでしょう。ここで言うところのHWPは、ITRONやT-Engineです。

Modeling does the future with the state-of-the-art engineering tool. ZIPC
組込みソフトウェアはどうなる?(図6)



SWPは、HWPが変化してもアプリケーションが影響を受けないように作成されます。HWPが性能重視に構築されるのに対して、SWPは機能やバリエーションを重視したプラットフォームになります。SWPでは、機能が部品として再利用できる環境や、現状のECLIPSEのフィーチャーのようなコンポーネント管理が行われます。SWPは、エンタープライズ系と異なり、各ベンダーからではなく、各メーカから登場すると予測します。何故ならば、組込みシステムのドメイン知識を持つところが、最適なSWPを構築できるからです。

キヤノンによる、MEAP (Multifunctional Embedded Application Platform) といったドキュメントやイメージをドメインしたプラットフォームがあります。自動車ドメインではJasPar (Japan Automotive Software Platform Architecture) という団体が設立されています。ロボットドメインでは、産業総合研究所のOpenHRP (Open Architecture Humanoid Robotics Platform) やNECのRoboStudioがあります。

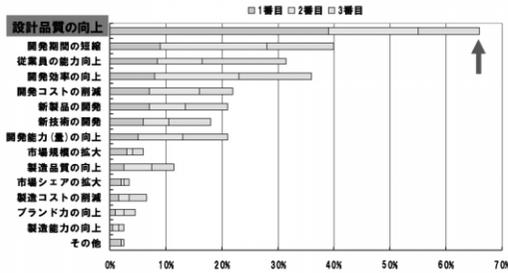
組込みソフトウェアに課せられている課題

今後おこると思われるプラットフォーム戦争は、別の機会に議論するとして、3層構造から構成される組込みソフトウェアは、依然として、

性能、バリエーションは、巨大・複雑化し続け、そして高性能への要求は高まるでしょう。

QCDについては、今のままの開発スタイルでは、破綻となります。特に組み込みソフトウェアでは、設計品質の向上(Q)が、開発期間の短縮(D)、開発コストの削減(C)を大きく引き離して、組み込みソフトウェアに課せられている課題として、経営者、事業責任者が考えていることが、経産省の組み込みソフトウェア産業実態調査(2006年版)からわかります(図7)。

Modeling does the future with the state-of-the-art engineering tool. ZIPC
組み込みソフトウェアに課せられている課題(図7)



2006年版 組み込みソフトウェア産業実態調査:経営者・事業責任者向け調査

Copyright 2006 CATS Co.,Ltd. All rights reserved.

CASEツールはどうか

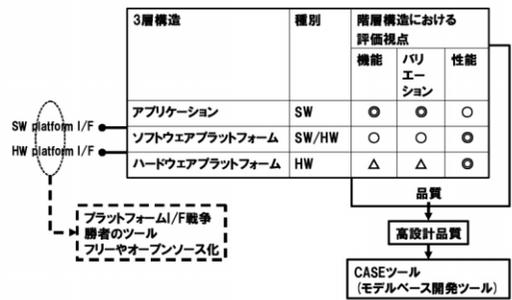
高設計品質には、図8に示すようにモデルベース開発(MBD)、またはモデル駆動型開発(MDD)が2020年では本格的に活用されることとなります。現在、モデリングが使えるかどうかの議論は終わり、モデリングをどう適用すべきかが議論の対象になっています。

経産省の組み込みソフトウェア産業実態調査(2006年版)からプロジェクト責任者は、モデリング手法導入効果で8割を超えて設計品質が改善されたとの回答があります(図9)。さらに、10万行以上では、分析・設計結果の記述手段としてのモデリング手法として状態遷移表が約5割を占めています(図10)。

MVCアーキテクチャモデル

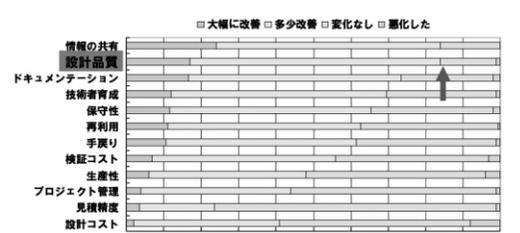
機能モデルはMVC(Model - View - Control)のそれぞれのモデルに分解されるべきです(図11)。UIの複雑・巨大化はDrawrialのような画面遷移モデルで記述され、Model(データ)構造の複雑・巨大化はEncirqやLinterといった組み込みDB上でのモデルで表現され、

Modeling does the future with the state-of-the-art engineering tool. ZIPC
CASEツールはどうか(図8)



Copyright 2006 CATS Co.,Ltd. All rights reserved.

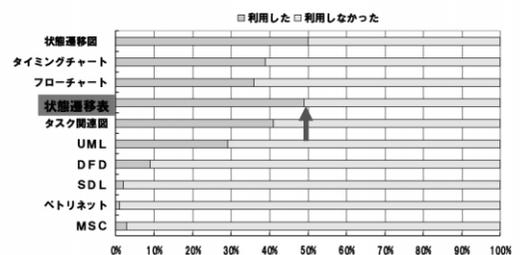
Modeling does the future with the state-of-the-art engineering tool. ZIPC
モデリング手法の導入効果(図9)



2006年版 組み込みソフトウェア産業実態調査:プロジェクト責任者向け調査

Copyright 2006 CATS Co.,Ltd. All rights reserved.

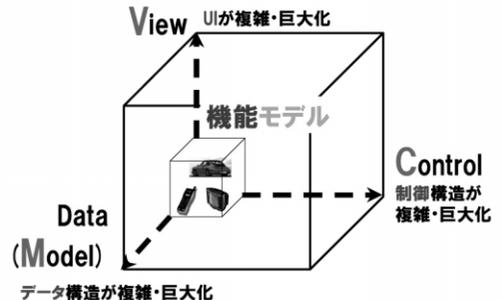
Modeling does the future with the state-of-the-art engineering tool. ZIPC
「分析・設計結果の記述手段」としてのモデリング手法の利用状況:10万行以上(図10)



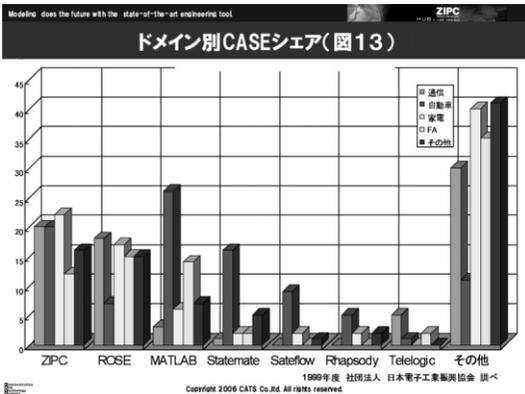
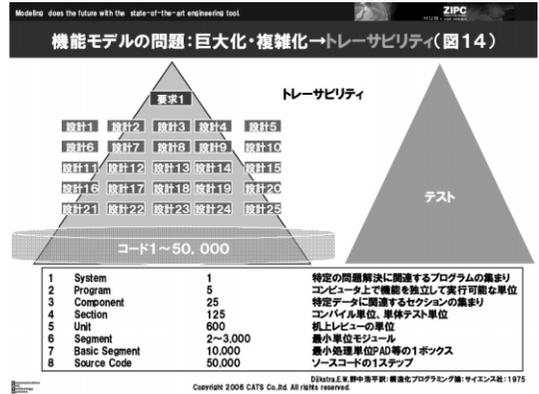
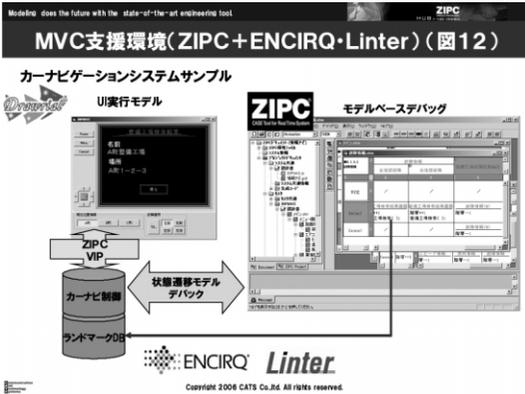
2006年版 組み込みソフトウェア産業実態調査:プロジェクト責任者向け調査

Copyright 2006 CATS Co.,Ltd. All rights reserved.

Modeling does the future with the state-of-the-art engineering tool. ZIPC
複雑・巨大化する機能の3軸(図11)



Copyright 2006 CATS Co.,Ltd. All rights reserved.



Controlの複雑・巨大化はZIPCのような状態遷移モデルで記述されることで、システムレベルの機能モデルアーキテクチャが構築できます(図12)。

MVCアーキテクチャパターンの要素ごとにCASEツールが用意されるべきです。そして、プラットフォームがドメインごとに用意されるように、CASEツールはドメイン毎にオプションが用意されるべきです(図13)。

ツールは道具ですから、ゴルフのクラブのように状況に応じての番手を揃えて、それを使いこなすことが求められます。

トレーサビリティ

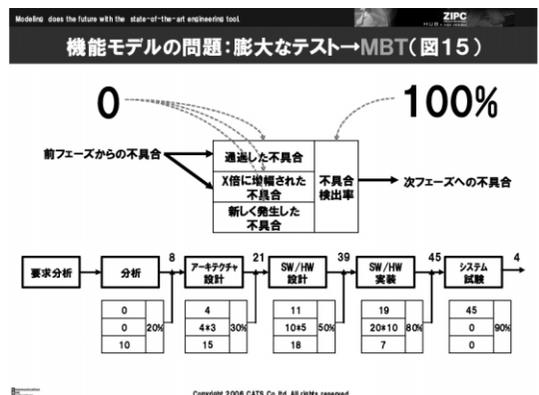
MVCアーキテクチャによって、作成されるデータモデル、ビューモデルそしてコントロールモデルは、さらに要求モデル、設計モデル、実装モデルと粒度の異なるモデルになります。そこで重要なことは、それぞれの工程間の成果物のトレーサビリティをとることです(図14)。

そのためCASEツールは、トレーサビリティを意識したリポジトリ構造が必要となります。トレーサビリティを意識していないCASEツールは、再利用を促進するプロダクトラインのようなプロセスを導入できなくなる危険性があります。

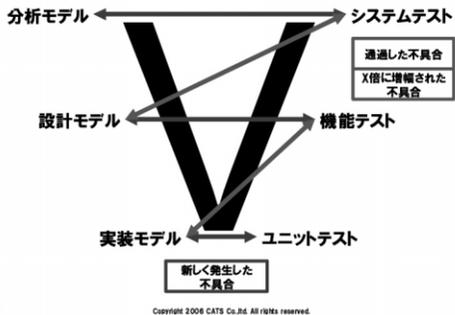
モデルベーステスト

膨大な機能を最終工程であるプログラムコードでテストを行う開発スタイルは、2020年には大きく改善されているでしょう。各工程粒度のモデルでテストが行われ、下流工程に上流工程の不具合を通過させないモデルベーステスト(MBT)が実現されているべきです(図15)。

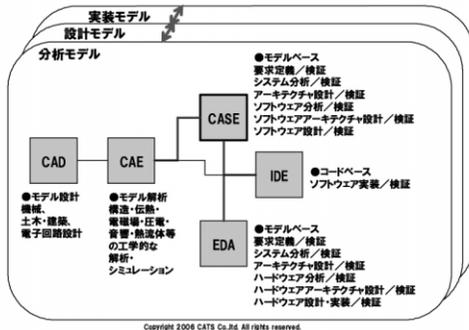
MBTは、図16の水平ラインで示される正当性確認テストと、斜めで示される検証テストがあります。正当性確認テストでは、現在行われている工程のモデルが正しい動作を行うかをテストすることで、現工程での不具合をなくします。



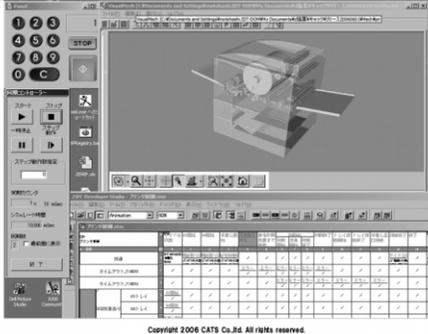
Modeling does the future with the state-of-the-art engineering tool. ZIPC
通過させないテストと発生させないテスト(図16)



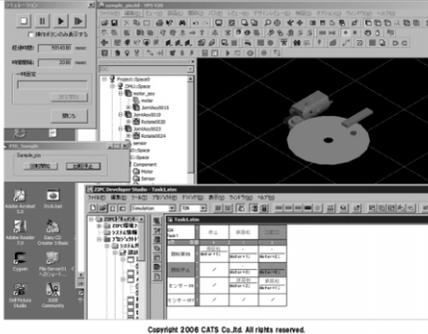
Modeling does the future with the state-of-the-art engineering tool. ZIPC
システムモデルテスト支援環境(図17)



Modeling does the future with the state-of-the-art engineering tool. ZIPC
CASE+CAE支援環境(ZIPC+VisualMech)(図18)



Modeling does the future with the state-of-the-art engineering tool. ZIPC
CASE+CAE支援環境(ZIPC+VPS)(図19)



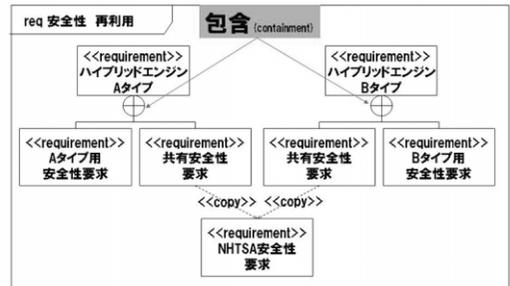
斜めラインの検証テストでは、1つ上流のテストを実施することで、現工程のモデルが、上流モデルの動作と異っていない等価性を検証します。さらに、上流工程のテストを再度行うことで、不具合の通過をなくします。

組み込みシステムの場合、組み込みソフトウェアモデルだけが動作する環境だけでは、システムレベルのテストができません。このため、将来は、CASE,CAE,EDA,IDEのツールは、それぞれのツールの粒度モデルで接続されるべきです(図17)。現在ZIPCでは、IDEやCAEとは接続されています(図18、19)。

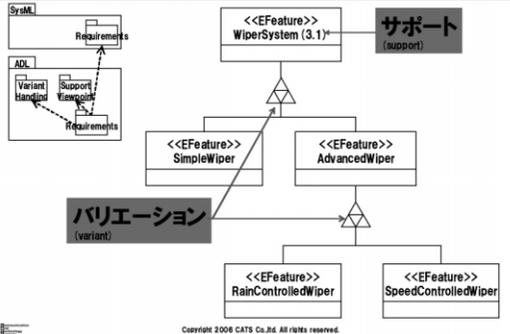
バリエーションモデル

バリエーションの巨大・複雑化(図4)をどのようにモデリングするかが重要なテーマです。SysML(図20)、ADL(図21)、フィーチャー図(図22)は、バリエーションがこんなに複雑だということを表記できます。2020年には、簡易化技法を用いて、もっとスッキリと記述できるようにしなければなりません。

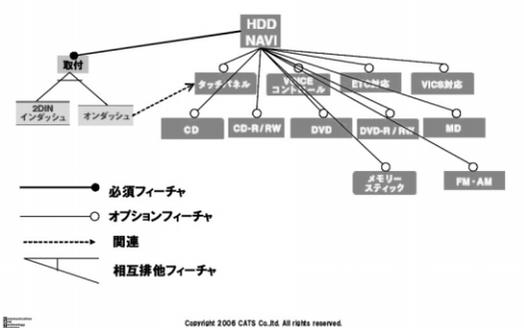
Modeling does the future with the state-of-the-art engineering tool. ZIPC
バリエーションモデルの問題:巨大・複雑化→SysML・要求図(図20)



Modeling does the future with the state-of-the-art engineering tool. ZIPC
バリエーションモデルの問題:巨大・複雑化→ADL(図21)



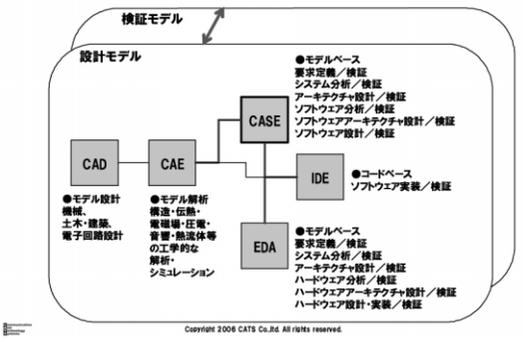
Modeling does the future with the state-of-the-art engineering tool
ZIPC
バリエーションモデルの問題: 巨大・複雑化→フィーチャー図(図22)



性能モデル設計・検証

性能モデルが設計、検証されない問題(図5)を、機能モデルから自動的に性能検証モデルを生成することで、解決します。モデル検査技法を設計ツールに内蔵し、裏方とすることで、組み込みソフトウェア技術者は、特定の知識なしで、モデル検査による自動検証の恩恵が得られるようになるべきです(図23)。

Modeling does the future with the state-of-the-art engineering tool
ZIPC
性能モデル: 性能モデル設計・検証ができていない(図23)



キャッツと九州大学とFLEETSで、ZIPC-REDを設計モデルとし、UPPAALの時間付きオートマトンモデルを性能検証モデルとして自動変換し、CTL式を自動生成し、リソーススケジューリング性能の設計、検証する研究が行われています(図24)。

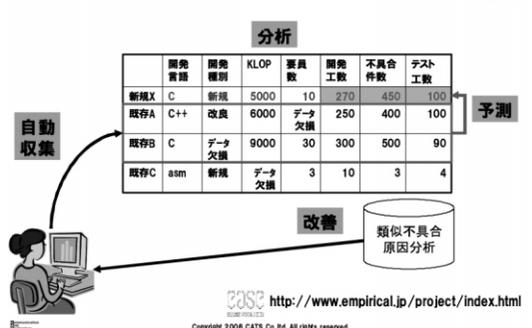
エンピリカル

機能、バリエーション、そして性能の設計品質をどれだけ高めたかを数値的に把握するには、エンピリカルソフトウェア工学が必要です。エンピリカルソフトウェア工学とは、ソフトウェ

Modeling does the future with the state-of-the-art engineering tool
ZIPC
性能モデル設計検証にモデル検査を適用(図24)



Modeling does the future with the state-of-the-art engineering tool
ZIPC
品質: 計る・測る・量るそして予測する(図25)



アのアクティビティを定量化し、統計的に分析を行い、そこから予測、改善を行うことです。

図25に示すように、2020年のCASEツールは、組み込みエンジニアの作業を自動収集します。自動的に収集された過去のプロジェクトデータから、類似性の高いプロジェクトを抽出し、傾向を分析し、開発工数や不具合などを予測し、どのように改善すべきかを過去の統計から助言します。

おわりに

2020年、ソフトウェアに不具合はつきものという世界常識に、日本の組み込みソフトウェアから不具合のない高品質なソフトウェアが出現し、再びMADE IN JAPAN神話が復活する。そして、その背景にはソフトウェア産業革命といわれるCASEツールによる自動化技術が成熟していることであろうと楽しく想像し、『超先』しています。

