・***Safe & Secure Systems & Software Symposium June, 2010***
http://www.azimuth-corp.com/conference/S52010/

(Air Force Research Labs (AFRL), National Science Foundation (NSF), Defense Research Project Agency (DARPA), NASA **の主要研究者が参加するカンファレンス**)

①エンジニアリングの基本的なアプローチは1960年から変わっていない

Making DARPA META Goals Come True: How do we Revolutionize Verification and Validation for Complex Systems?

Dr. Kirstie Bellman, Aerospace Integration Science Center (AISC) The Aerospace Corporation

(http://www.azimuth-corp.com/conference/S52010/papers/presentations/Day_3/Making%20DARPA%20META%20Goals%20Come%20True%20-%20How%20do%20we%20Revolutionize%20Verification%20and%20Validation%20for%20Complex%20Systems%20[Bellman].pdf)

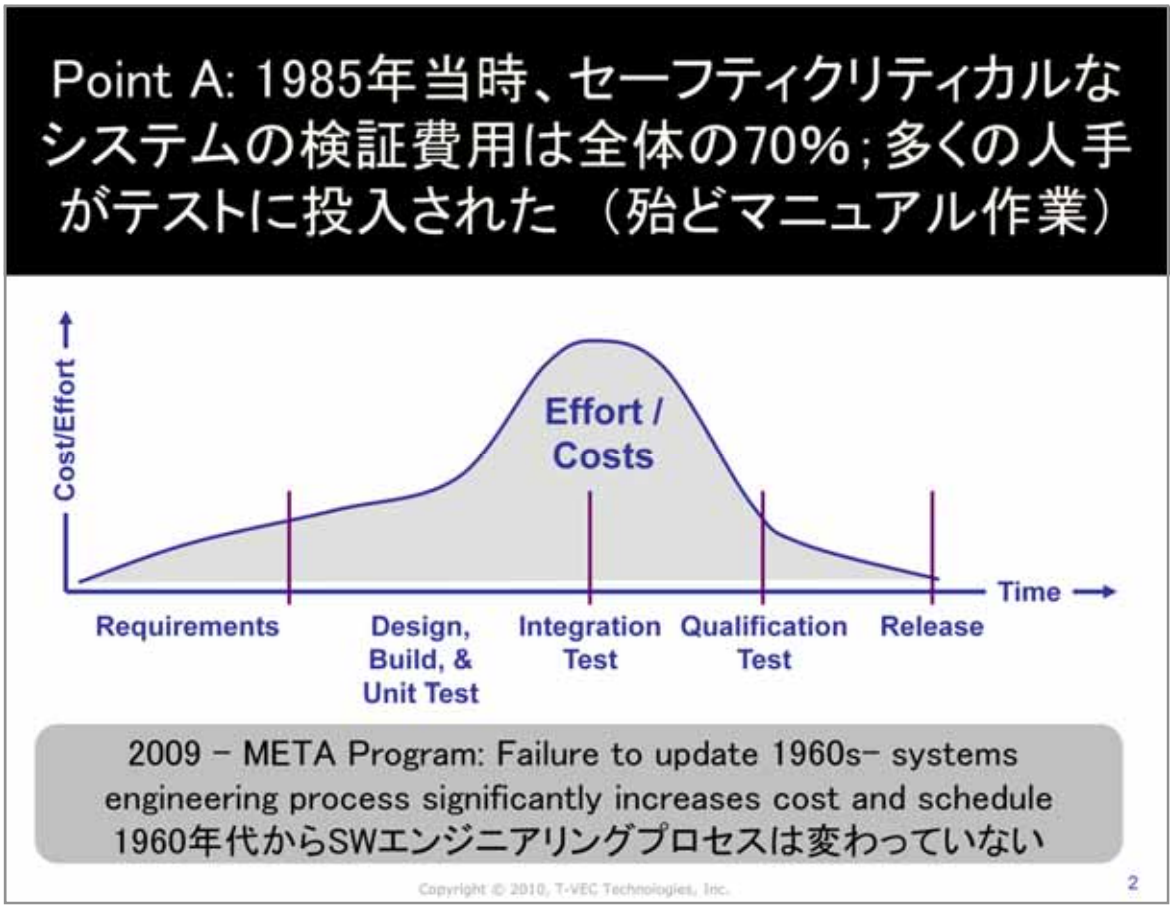
②発表者であるMark R Blackburnは、25年間セーフティクリティカルなシステムの開発に関わってきた (FAA DO-178A、DO-178B 認証、FDA関連システムへの取り組みなど)

③V&V 作業の多くを自動化支援する仕組みを考案し創出

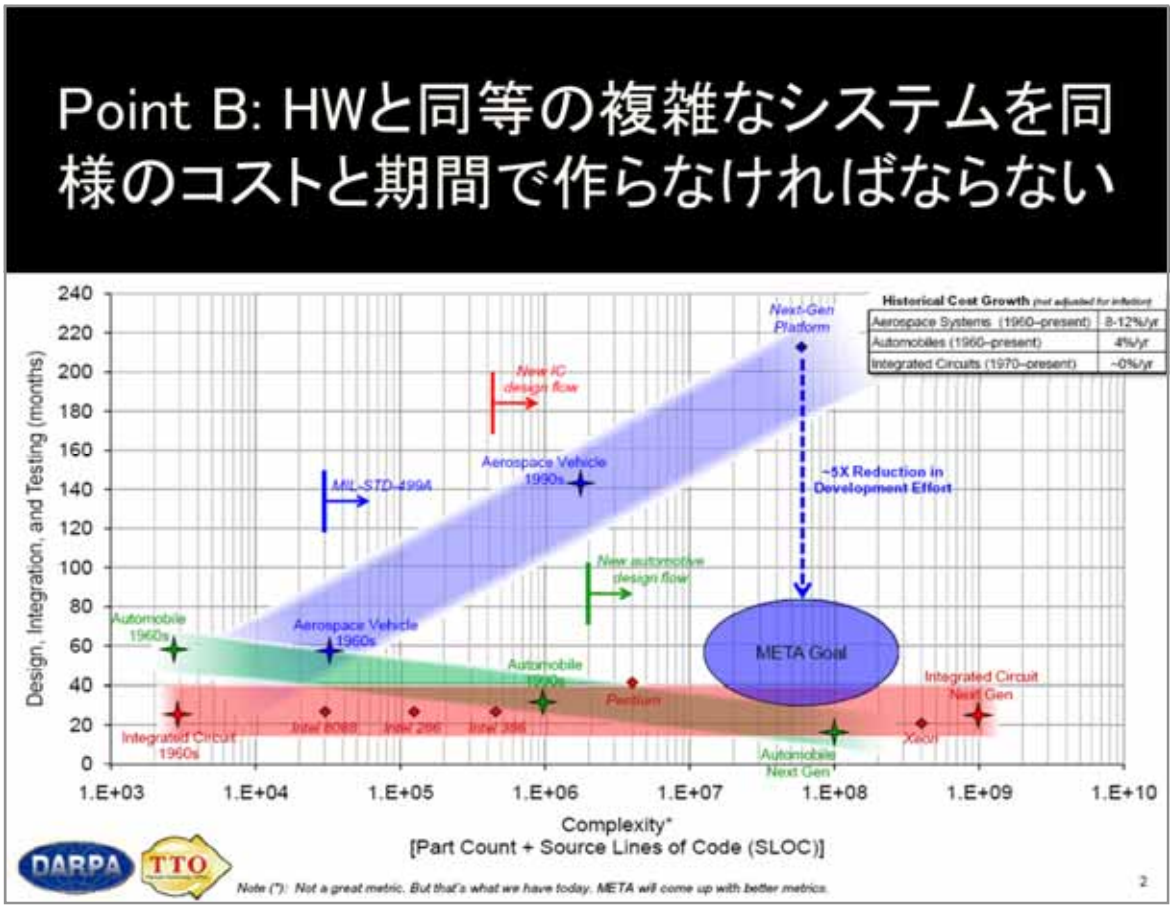
- **形式手法による定理証明であり副産物としてテストベクタを生成できる**

④セーフティクリティカル、ミッションクリティカル、商用システムに活用されてきているが、まだ広く知られるに至っていない

⑤META プログラムで紹介された課題克服に活用できる特徴を紹介する



1. 1985**年に**FAA**の認証を取得した時の例**
2. **このセーフティクリティカルなシステムの**70%**の工数は検証作業**
3. **検証のエビデンスを得るためにプロジェクト後期に多くの人が投入される**
4. **殆どのテストはマニュアル（人手に頼った作業）**



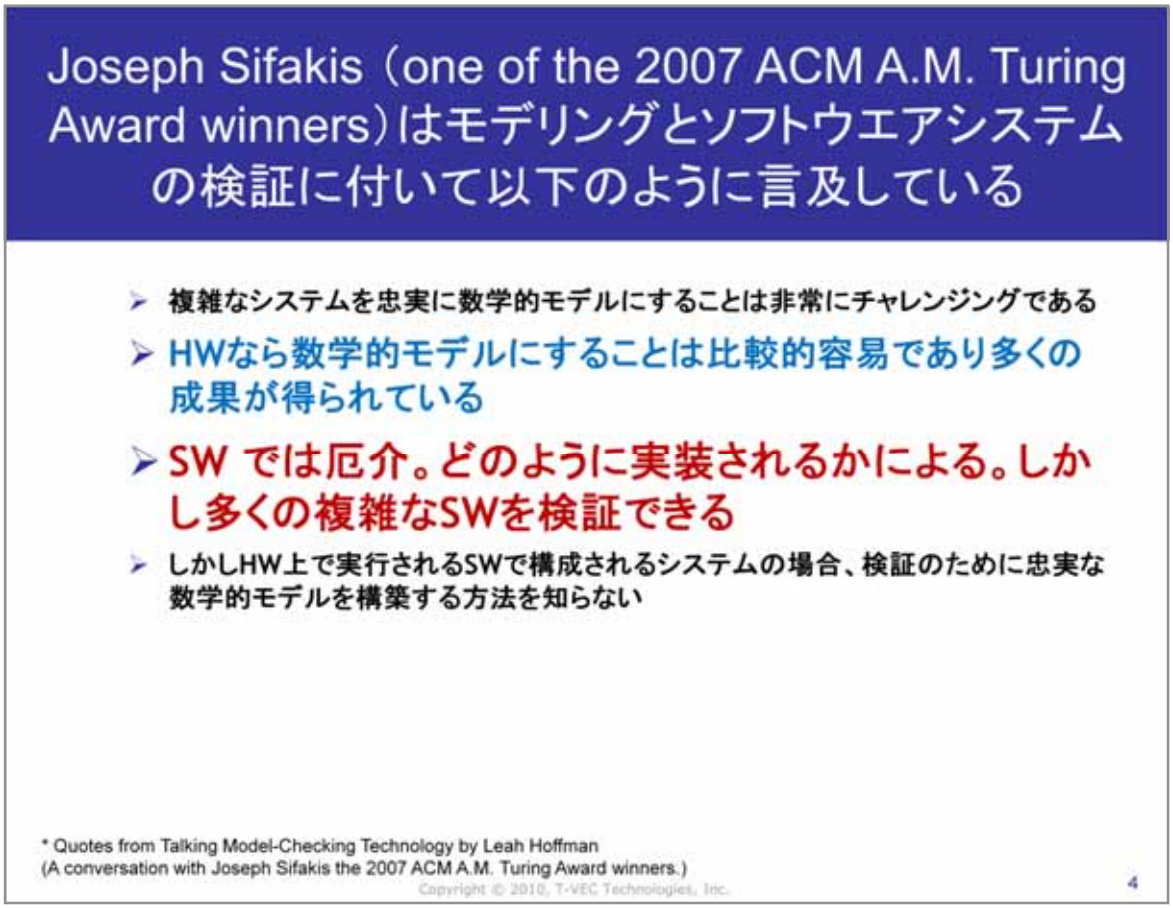
目的はより進んだエンジニアリングツールの応用。IC（集積回路）開発で行われているのと同様の、より厳密で建設的なアプローチをSWエンジニアリングプロセスに持ち込むこと

Image credit: DARPA META program *APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED*

V&V cost and Certification

For FAA compliant DO-178B Level A software, the industry usually spends 7 times as much on verification (reviews, analysis, test). So that's about 12% for development and 88% for verification.Level B reduces the verification cost by approximately 15%. The mix is then 25% development, 75% verification.

Randall Fulton FAA Designated Engineering Representative

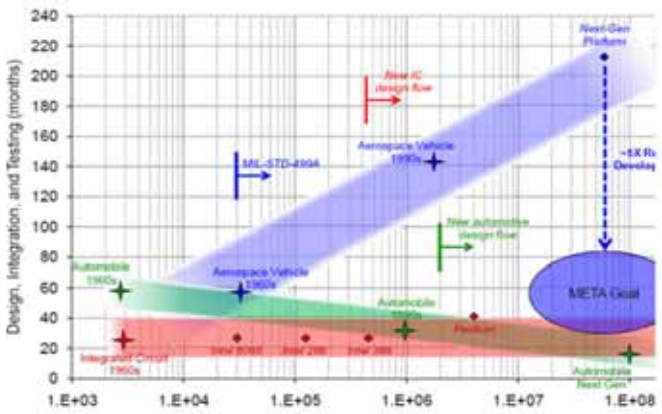


ここでキーポイントは、

1. ハードウエアを数学的モデル化することは比較的容易
2. ソフトウエアでは難しくなる。どのように記述されているかが問題になる

- 一般に仕様上のエラー削減は、実装に対するそれに比較してインパクトが大きい[Nancy Leveson].
- T-VECは仕様･設計モデルを形式手法のよって定理証明し、テストベクタを生成できる。それはリニア、ノンリニアをサポートし、フロートやダブルなどあらゆるデータ型に対応するので、実HWシステムレベルで仕様･設計と実装の一致性検証に活用できる。

Call to action: ソフトウエアシステムとハードウエア(IC)の違いに取組む必要がある

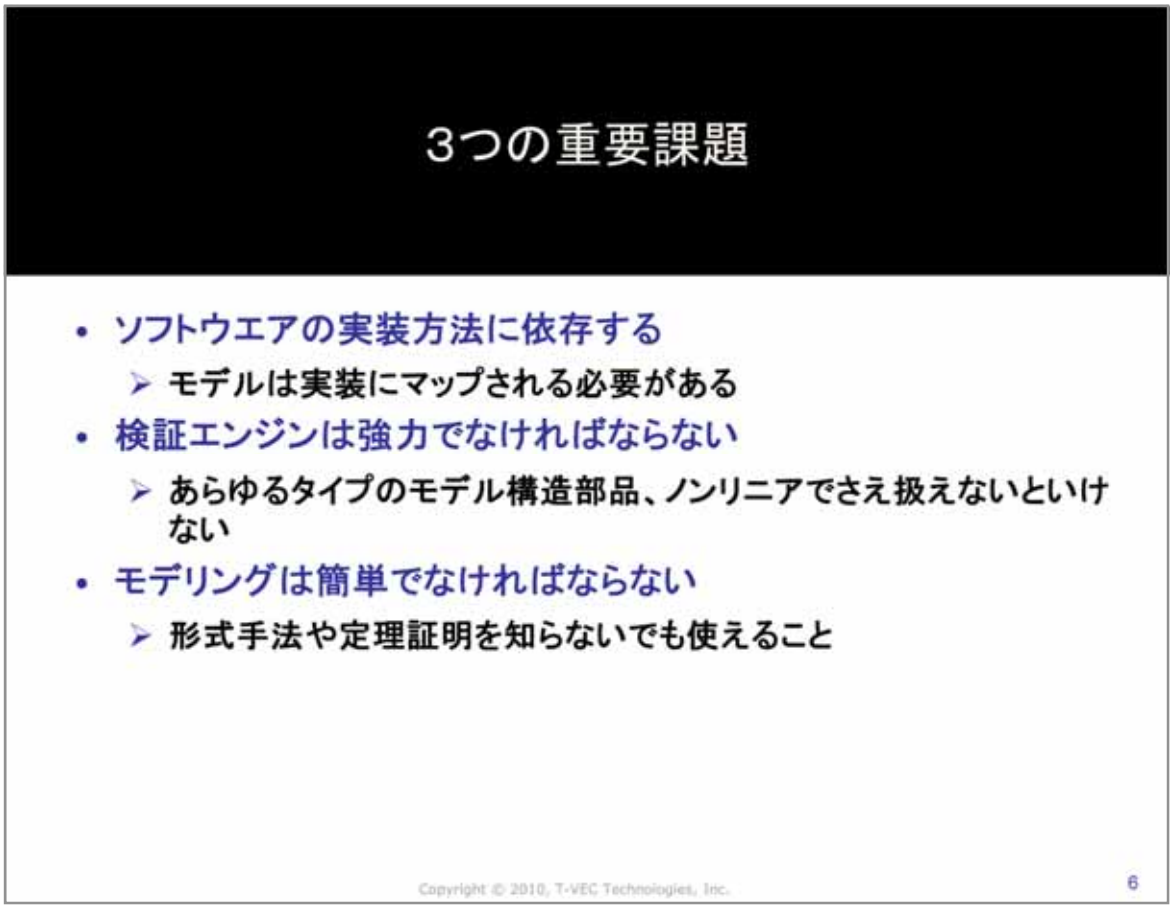


What's Different?

ソフトウエアの振舞いにはフローティングポイント型の変数に対するノンリニアな関係・コンストレインツがある

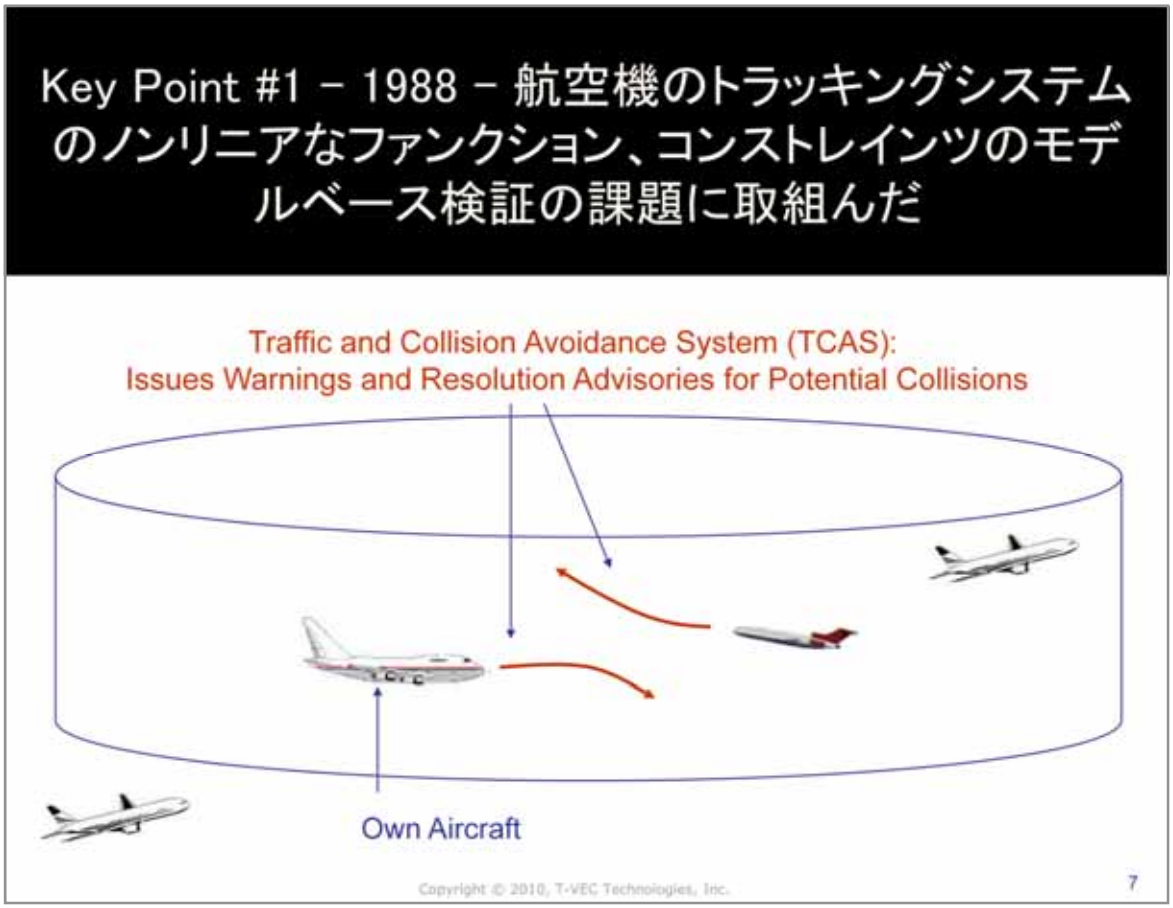
5

1. **ハードウエアとソフトウエアの多くの違いについて論ずる前に**
2. **基本的な違いのひとつを見てみたい。それはソフトウエアの振舞いの中で、フローティングポイント型の変数に対するノンリニアな関係・コンストレインツ　がHWとの違い**

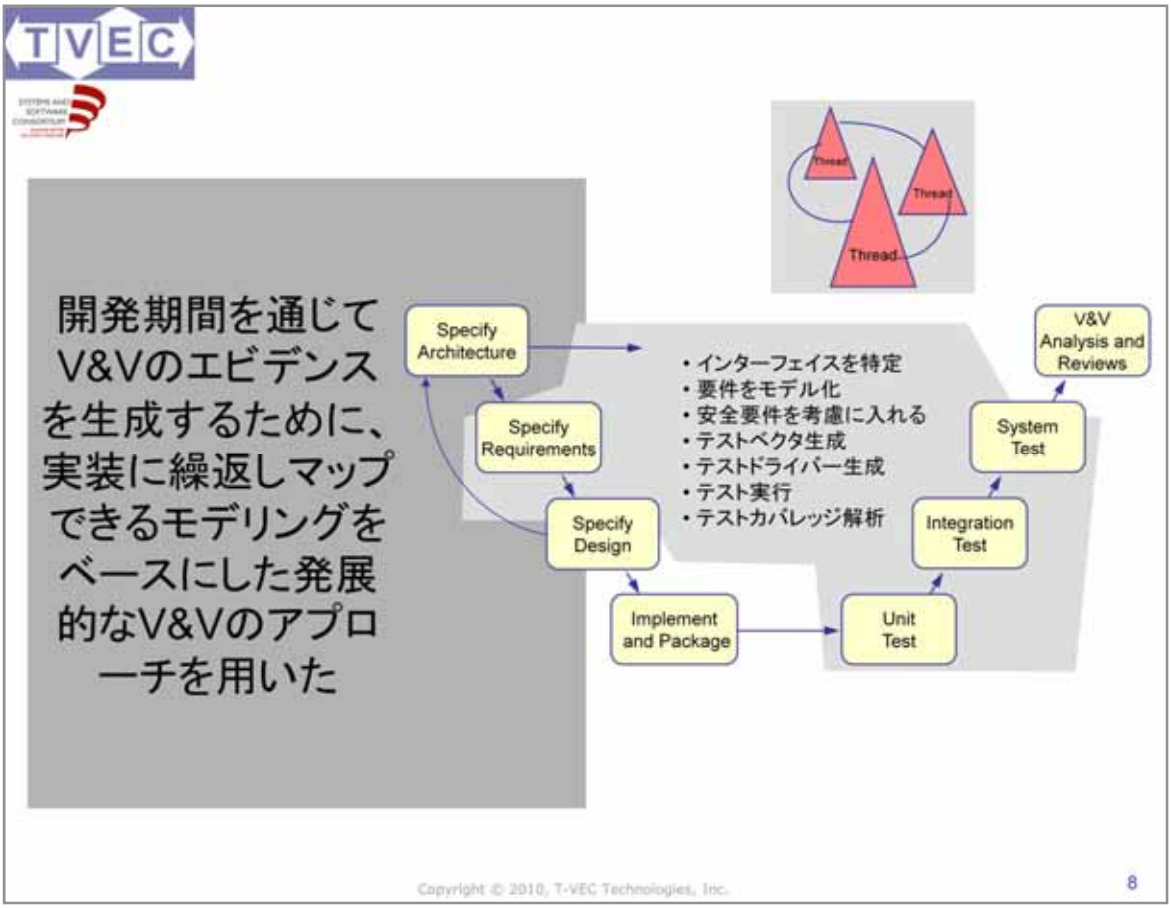


このプレゼンで紹介する3セクションには2つの横断するテーマがある

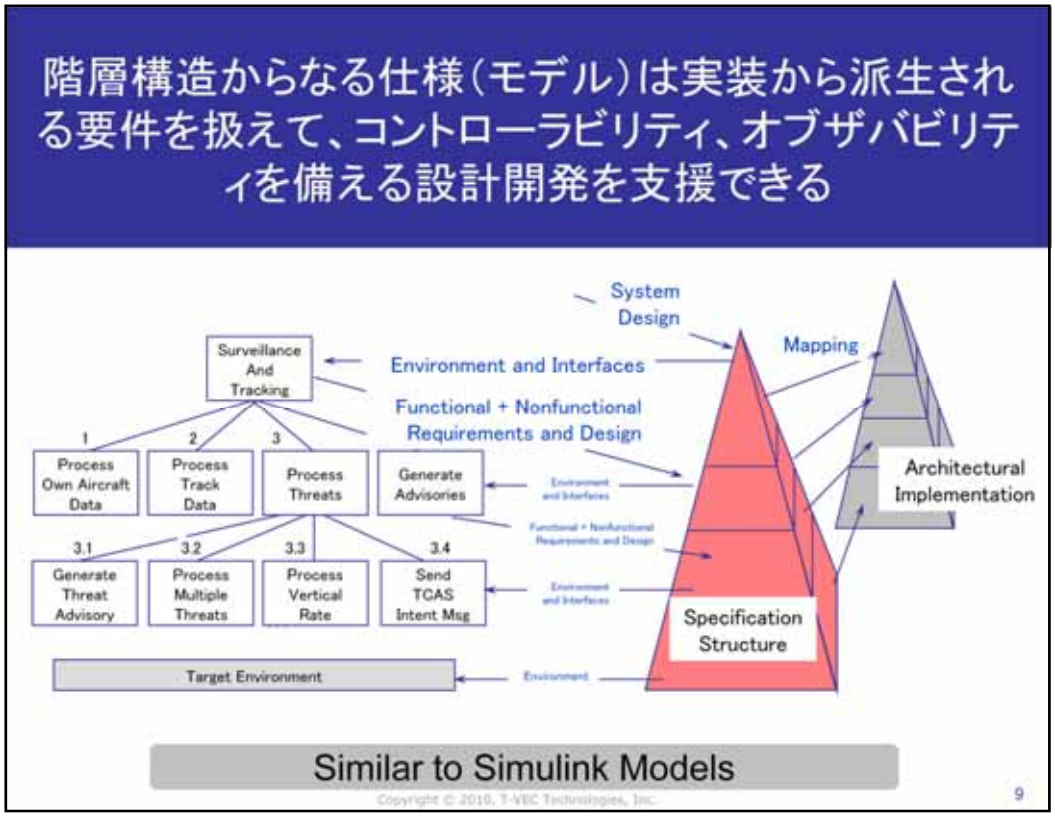
- ソフトウエアの記述方法によって自動検証への影響があること
- 実装にマップされるモデルの構造部品の全てのタイプを十分に扱えないような検証エンジンでは使い物にならないこと



1. 1988**年に検証に対する基本的かつ費用の掛かる問題に取組む機会を得た**
2. **体系的に支援できる手法とツールを決定する機会を与えられた**

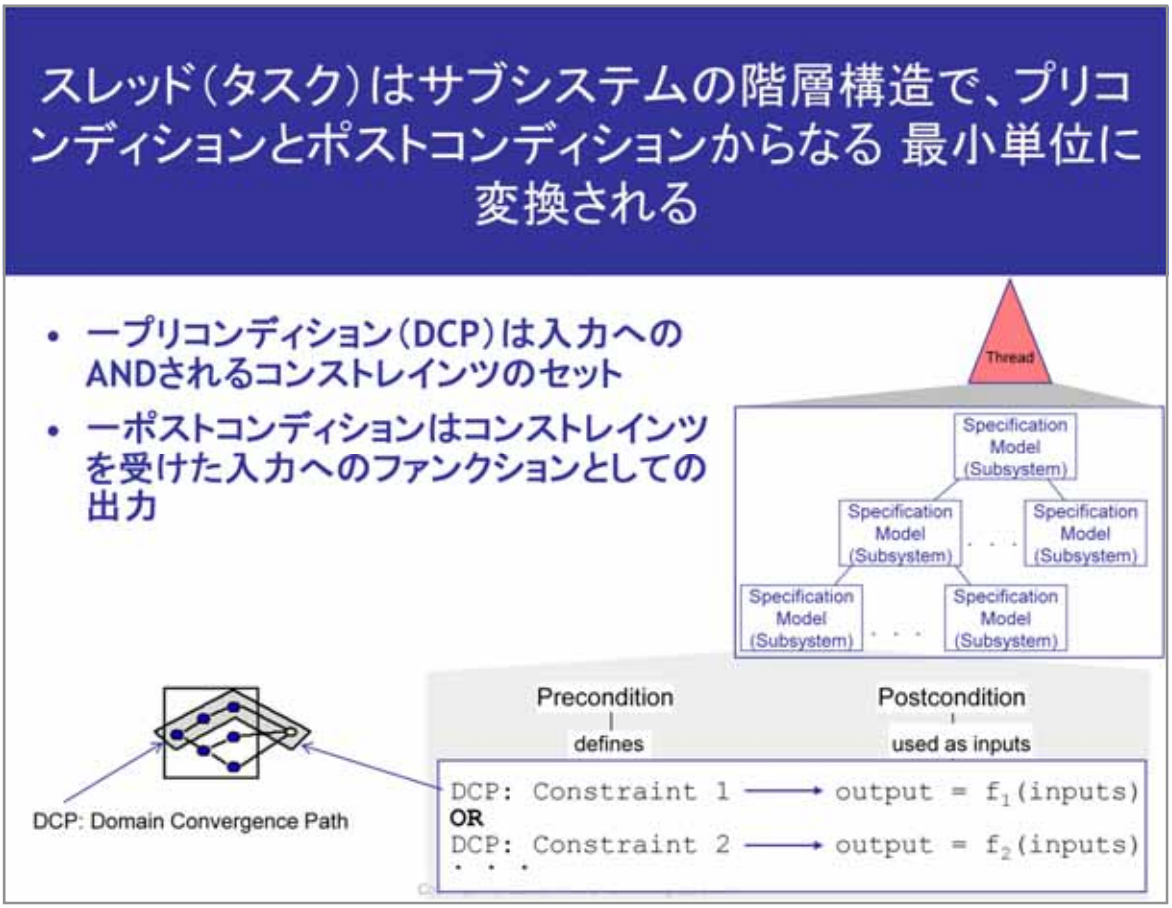


1. **このプロジェクトでFAAと密接にコンタクト。要求仕様ベースのテストが一つの目標であった**
2. **上流の要求仕様、実装から派生される要件、安全性のための要件など、全てに対して同様に従来からのVモデルを採用した**
3. **そしてソフトウエアシステムの全てのスレッド（タスク）に同じプロセスを採用**
4. **開発期間を通じて検証のエビデンスを生成することを目標として**

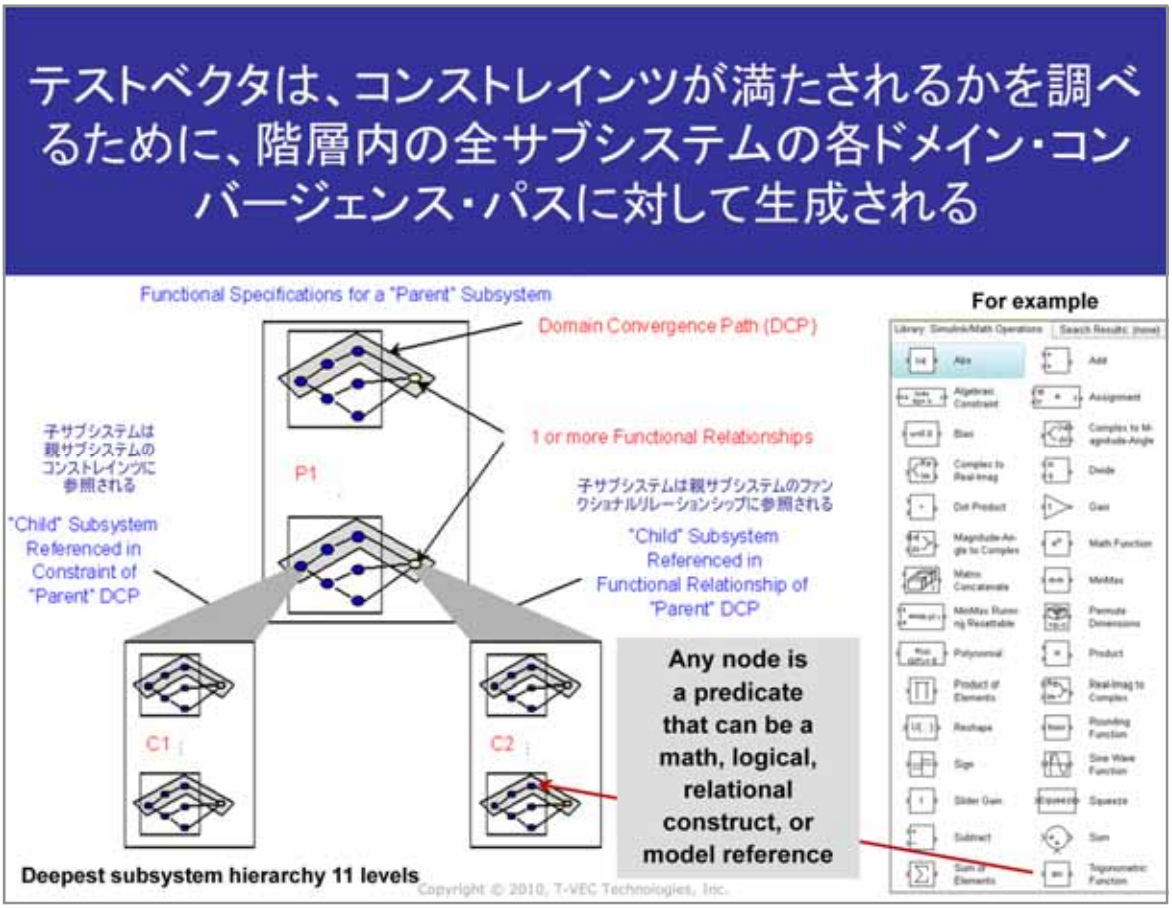


1. **実装にマップできる階層的な仕様（モデル）を創出**
2. **インターフェイスが次の要素をサポートすることを意識した。**

コントローラブル・制御可能（入力と状態の設定）、オブザバビリティ・可観測性（出力を得る）

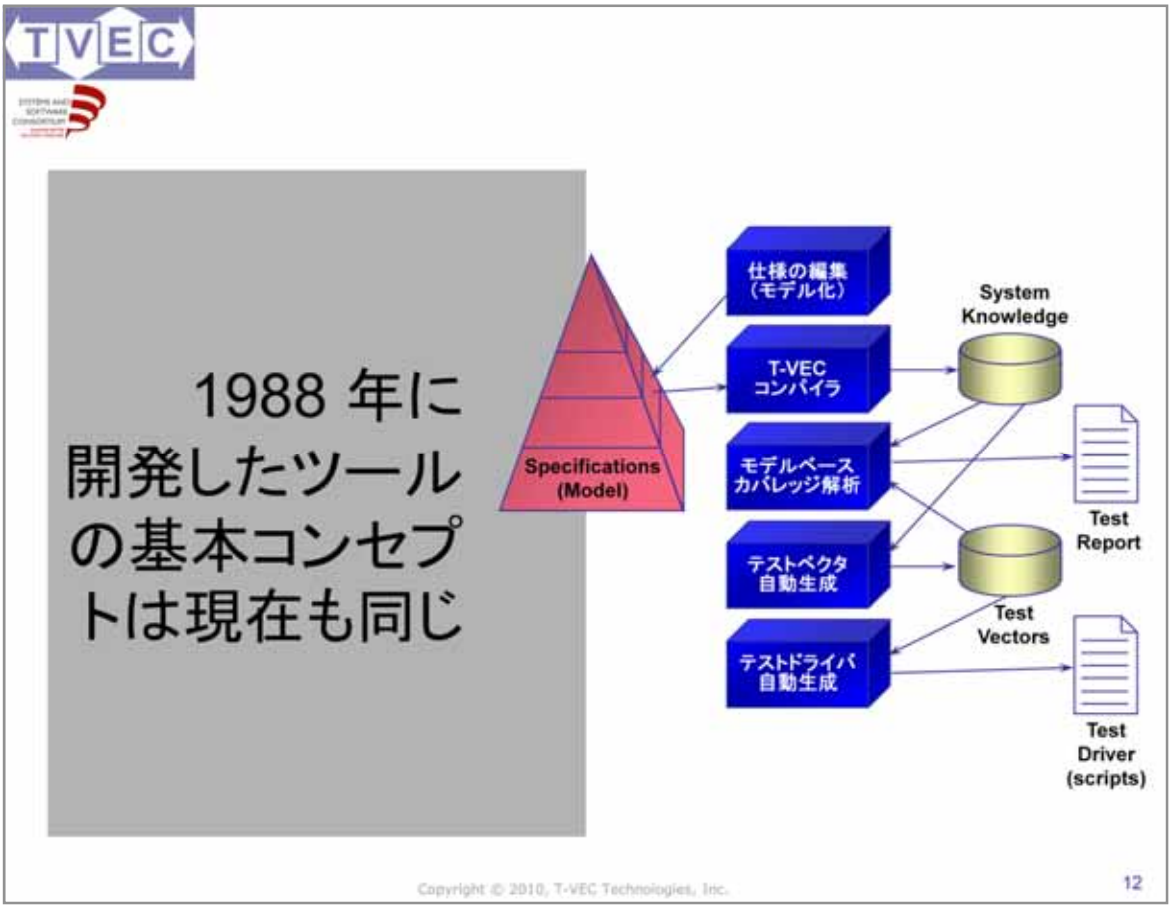


1. **全てのスレッド内の全サブシステムは**DCP（Domain Convergence Path）**というローレベルの形式に変換**
2. DCP**は本質的に、プリコンディション（入力とコンストレインツのセット）とポストコンディションとして表現されるファンクショナルリレーションシップ**

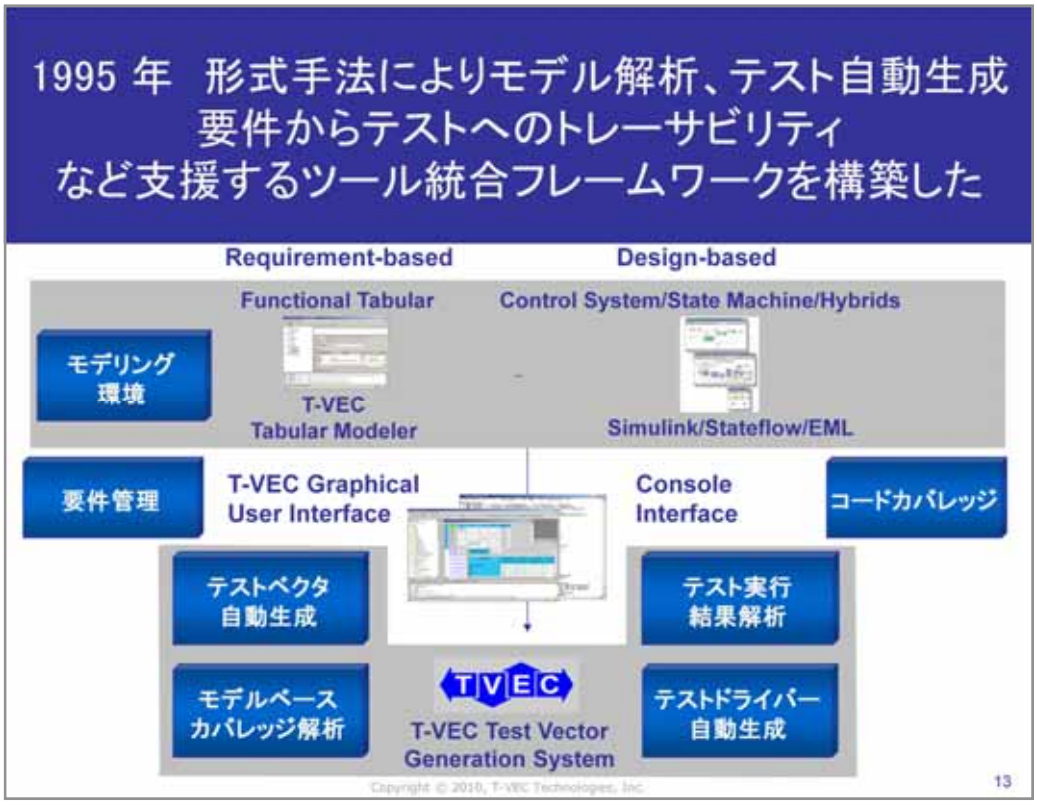


1. サブシステムの階層構造からなるシステムを形式手法で定理証明する仕組みの説明
2. 各サブシステムは1つ以上のDCPを持ち、その各ノード（コンストレインツあるいはファンクション）は本質的に述部として、他のサブシステムへのリファレンスとなる。
3. それゆえファンクションはコンストレインツとして、上流レベルのプリコンディションの一部となる
4. これら述部は標準的な数学、数理論理学、プログラム言語の論理オペレータなどで構成されるが、Simulinkのモデルライブラリ部品もサポートする
    - インテグレータ・積分器
    - N次のインターポレーションテーブル
5. T-VECはコンストレインツ（DCP）をベースにしてサブドメインの入力領域からテストデータを選択・抽出
6. DCPはMCDCテストカバレッジをサポートするために、相当するプログラムのパスコンディションにマップされる

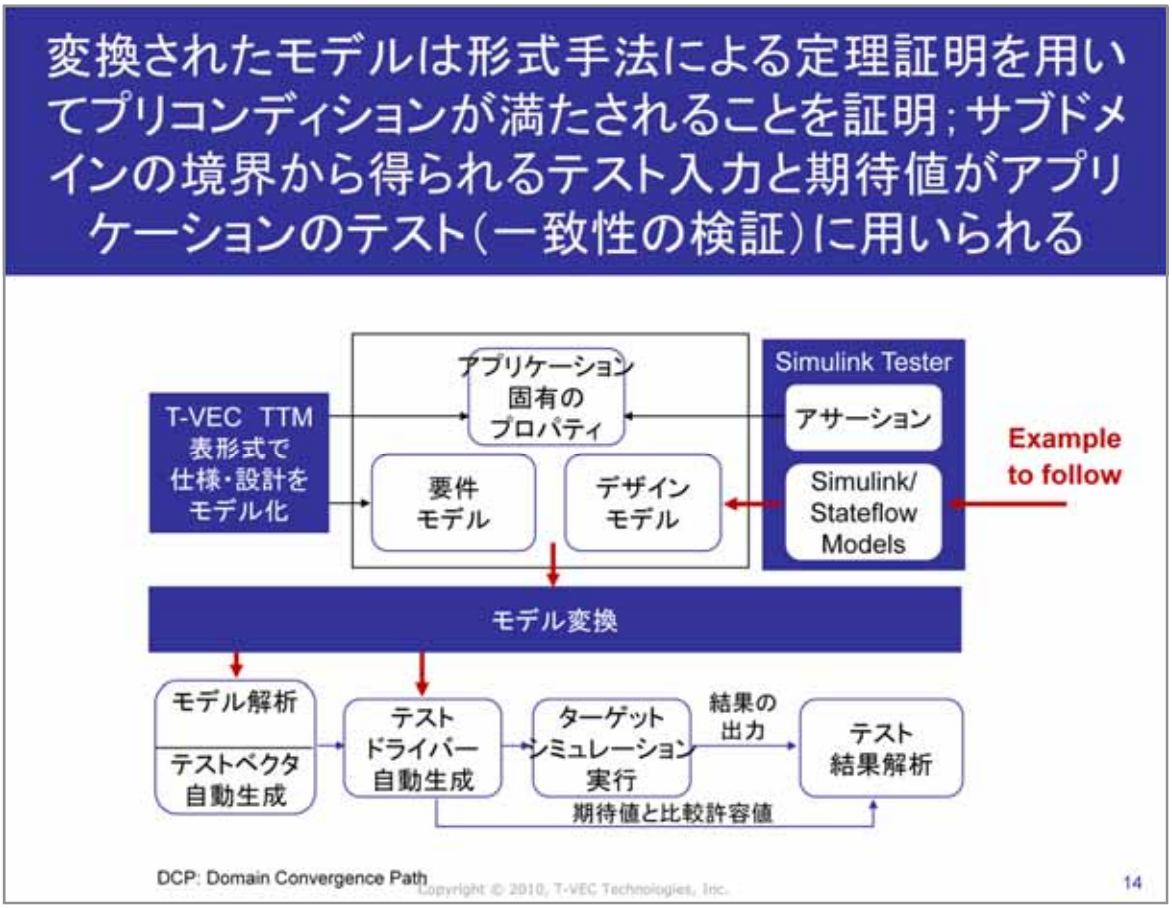
＊親側のコンストレインツを受ける変数の入力レンジは、下流のサブシステムのコンストレインツも受ける。そこで得られる入力領域内で期待値が取れるかを検査するテストベクタ生成を行い、サブドメインのDCPがシステム内で成立することを証明する（T-VECによる形式手法、定理証明）



1. 仕様記述のエディタ機能を拡張したことを除いて、ツールの基本コンセプトは現在のツールと同じ
2. 階層からなる仕様はローレベル (DCP) に変換されてSystem Knowledgeとして保存される
3. このSystem Knowledgeはテストベクタ生成のために使用される
4. モデルベースカバレッジ解析は、ずべてのDCPがテストベクタを持つことを解析する別のツール
    - テストベクタが持てない場合、モデル上に問題があるということ
    - この手の問題を、この後簡単に紹介する
5. 生成されたテストベクタを用いてテストドライバーを生成する機能がある

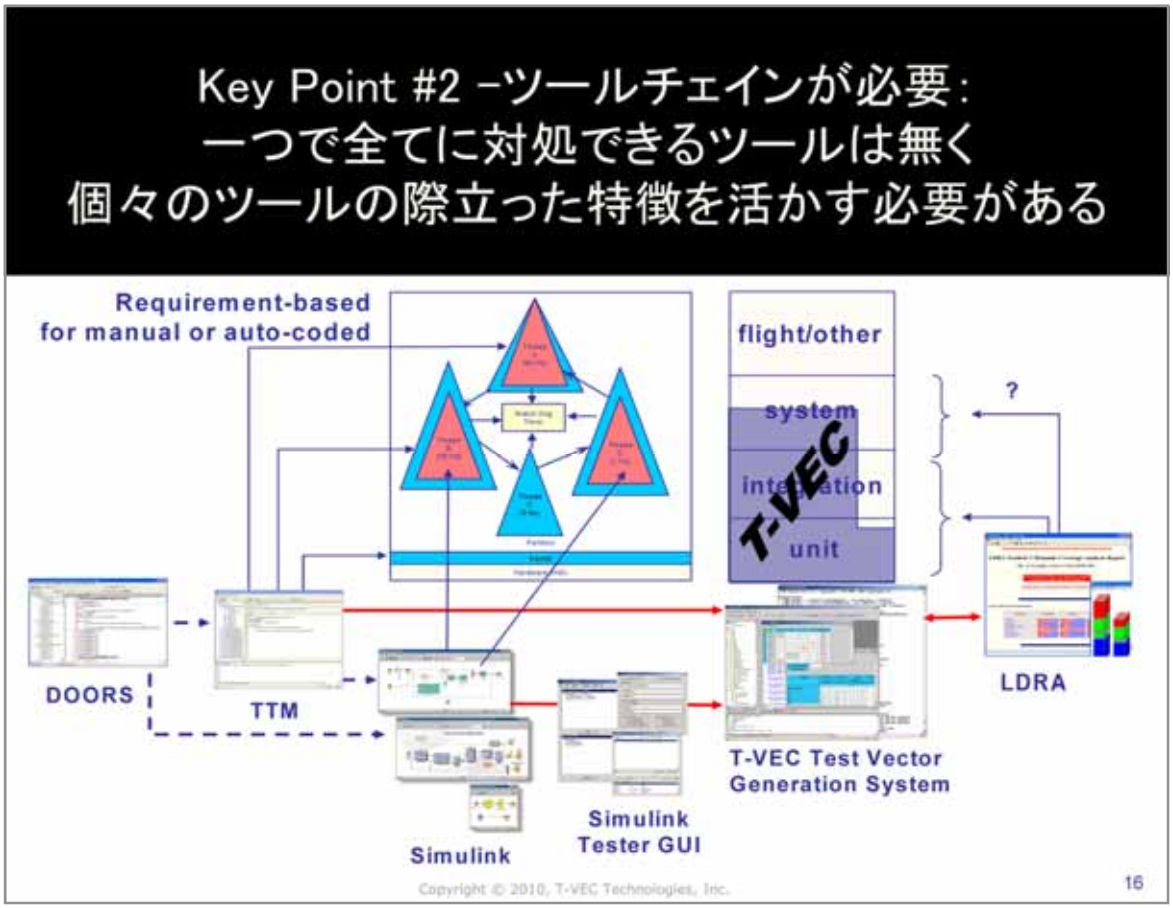


1. 1995**年から様々なモデルツールを**T-VEC**に変換するプロジェクトを手がけた**
2. Consortium Requirements Engineering Method (CoRE)**をサポートする**SCR
3. CoRE**の基である**Software Cost Reduction Method**をサポートする**SCR**ツールは** Naval Research Laboratory（**米・海軍研究所**）**で開発された**
4. F16**向けに**MATRIXx **にも対応**
5. **また**DOORS**といった要件管理ツールとの統合**、LDRA**社のコードレベルのテスト実行・カバレッジ解析機能との統合も、顧客の要求に合わせて行ってきた。**
6. T-VEC**の自動テストベクタ生成システムは、モデル解析、テストベクタ生成、要件からテストカバレッジに至る解析、テストドライバー生成を行えるので、検証・テストに於ける人手に頼って間違いの基になりやすい作業の多くを自動化できる。そして開発ライフサイクルに一貫したテストを支援して、市場投入までの時間を削減し、開発とメンテナンスのコストを軽減し、製品品質を向上させることに貢献する形式手法・定理証明のツール。**

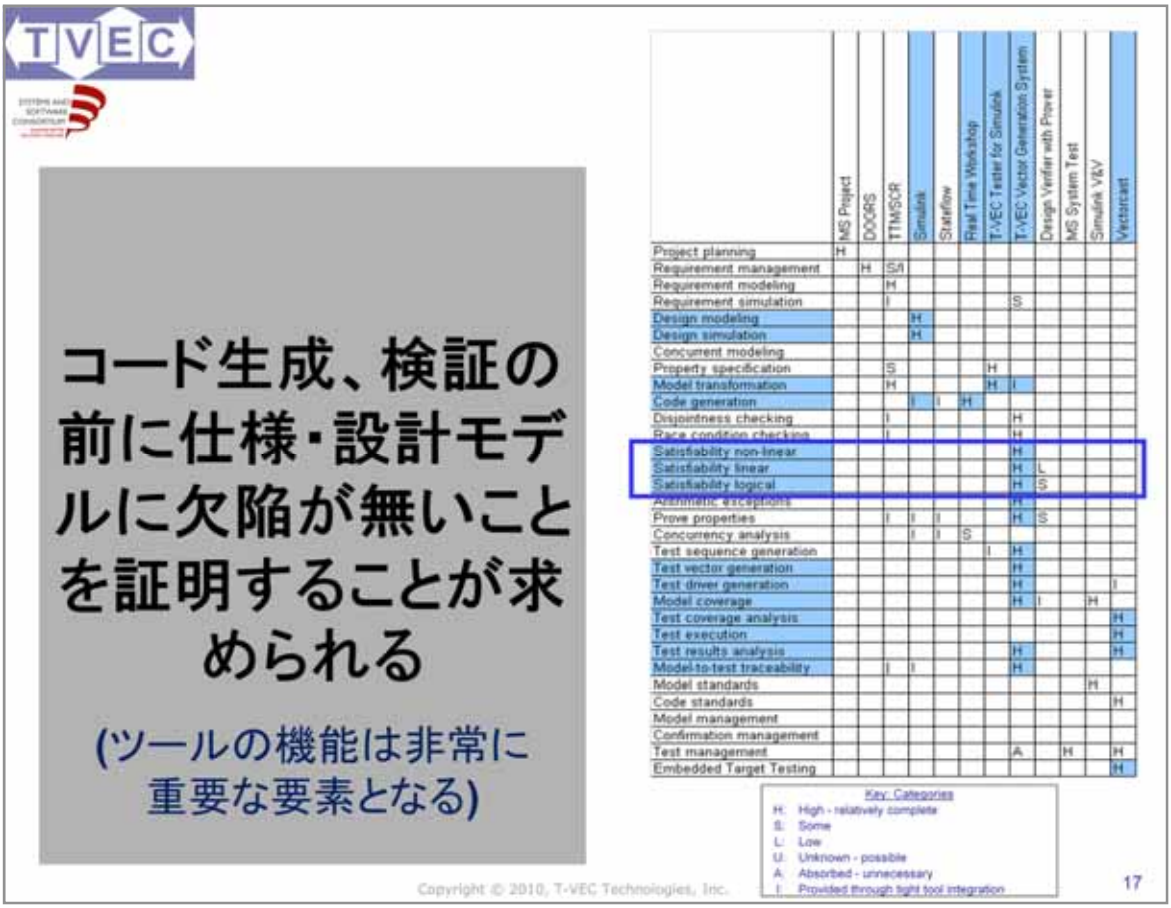


1. **モデルは要件、デザイン、アプリケーションの特性・プロパティ（安全特性など）を描写できる**
2. **それらはローレベルの形式（DCP）に変換される**
3. **モデル解析は形式手法による定理証明（事前条件を満たす入力の有無を証明する）**
4. **ベクタ生成機能がサブドメイン境界の入力値に対する期待値を計算**
5. **テストドライバー生成機能を用いてテストベクタは、あらゆる環境下で実行して結果を得るためのテストドライバーに変換**
6. **テストベクタ内の期待値をテスト結果として得られる実出力値と比較**

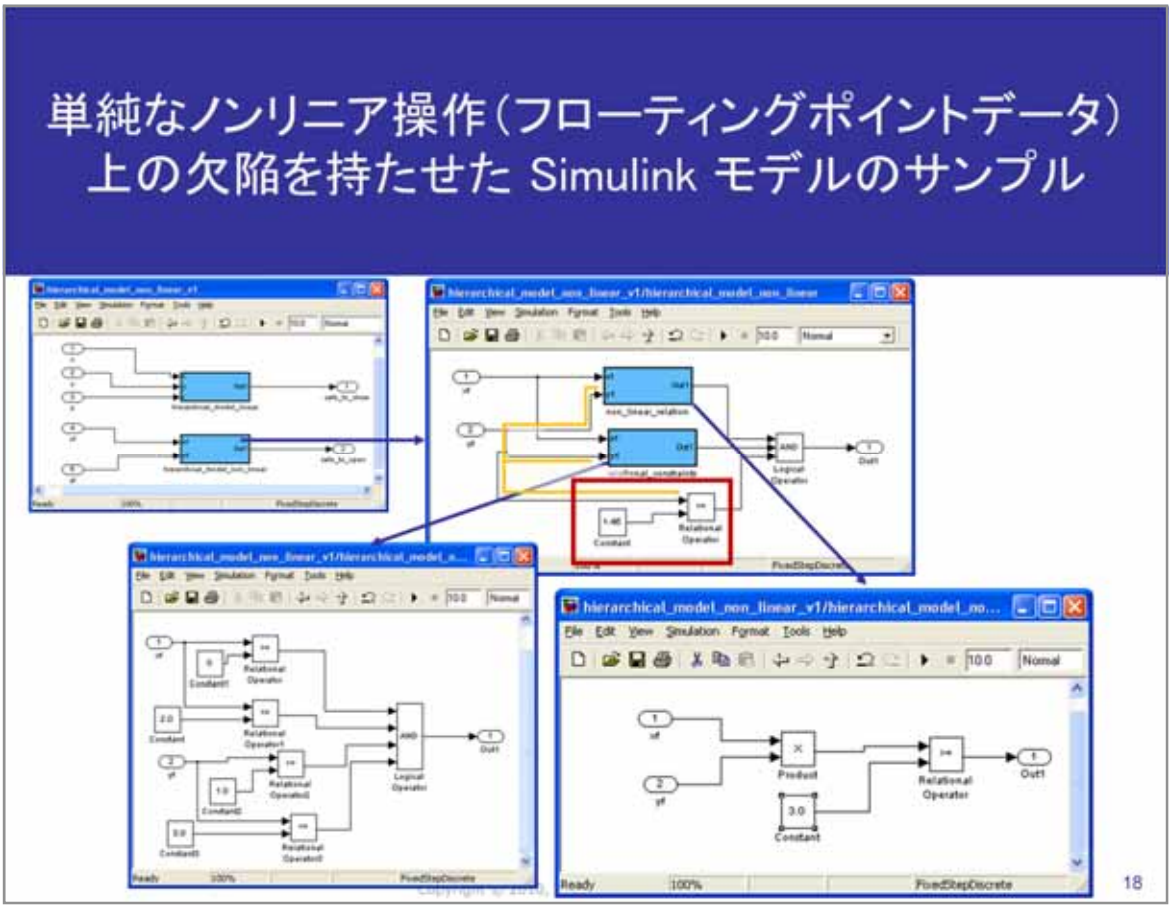
様々な言語環境、実行環境に合わせたテストドライバー生成に活用できるテンプレート（スキーマ）を提供



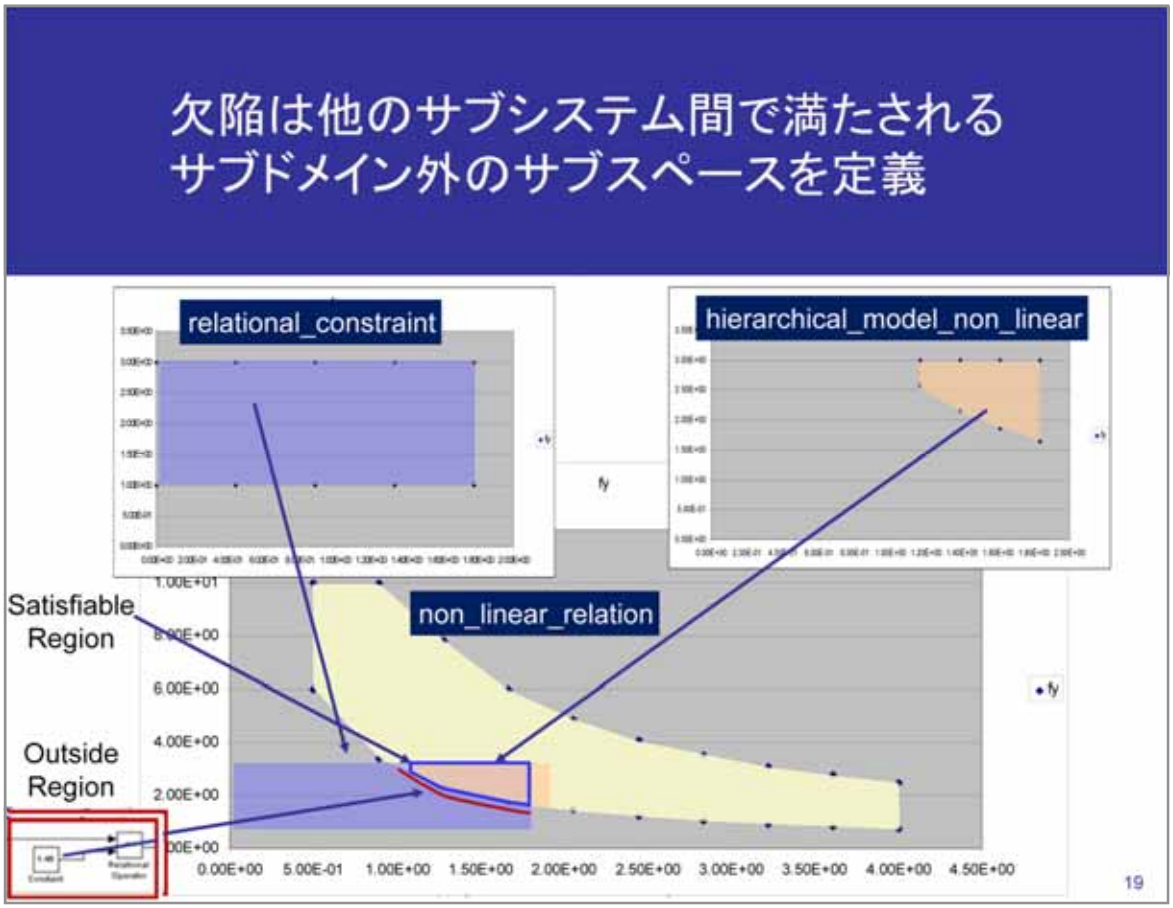
1. **テキストベースで管理される要件からテストに至るトレーサビリティを支援できる開発ライフサイクルに一貫したツールチェインが求められる**
2. **ライフサイクルの殆どの部分をサポートできる特別なツールが必要**
3. **コード生成が出来ることで制御系システムで**Simulink**はポピュラーとなっているが全システム内の**30%**を作っているにすぎない**
4. TTM**は**Simulink**以上にシステムの他の部分の機能性のモデル化に活用される**
5. **単体テスト、**HW**と**SW**の統合テスト、システムテストに活用できる**



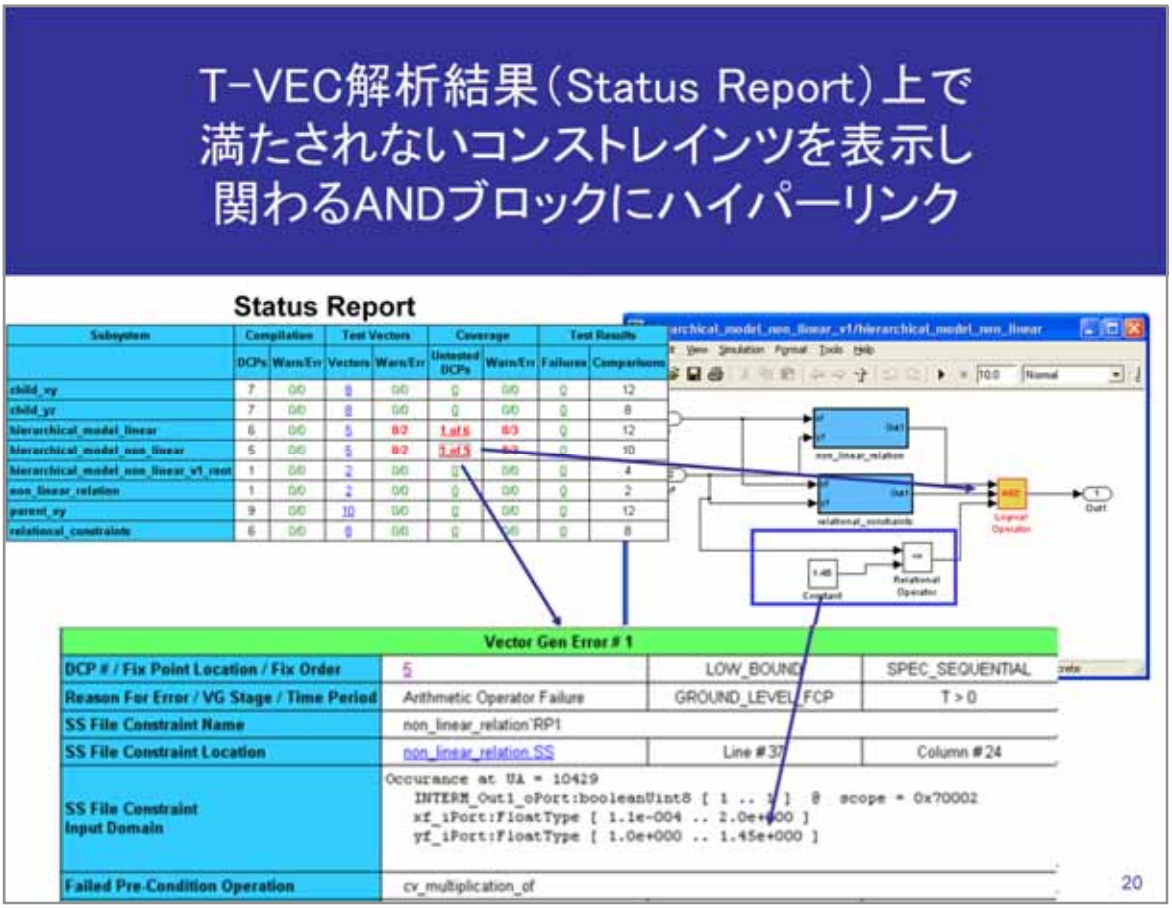
1. **ある企業からツールチェインの有効性に関しての調査依頼を受けた**
2. **モデル解析とテスト生成能力についての興味**
3. **モデル上のロジカル、リニア、ノンリニアなコンストレインツの解析能力に関して調査した**
4. **モデルが変換されてコードとして生成されるのなら、モデルに欠陥が無いことは重要**



1. **欠陥を含むモデルを作った**
    - **リニアなコンストレインツを満たさないものと、**
    - **ノンリニアなコンストレインツを満たさないもの**
2. **赤で囲まれたノンリニアなコンストレインツは下層サブシステム内の**Gain**ブロック（増幅器）に関係する**
    - **シンプルなノンリニアな関係であるが**
3. **次のスライドで詳細**



1. **3つのサブシステムごとで制限された領域**（constrained space）
2. **小さな赤色の線はリレーショナルオペレータのコンストレインツがオーバラップする領域から外にあることを示している**
3. **結果、全てのコンストレインツを満たす入力領域が存在しないことがわかる**



1. T-VECのステータスレポート上に、テストベクタを生成できないDCPが有る事が表示（リニアとノンリニアの両方）
2. それらはモデルの問題箇所にハイパーリンクできる
3. この例ではANDを満たせないということ

Mathworks 社 Design Verifier (DV) では 複数の満たされる対象にテストが生成できなかった (T-VECではテストベクタを生成)

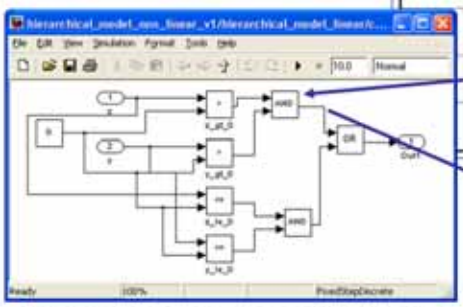
Design Verifier Report →

Number of Test Objectives: 124
Objectives Satisfied: 73
Objectives Proven Unsatisfiable: ..11
Objectives Producing Errors: 40

hierarchical_model_linear/child_yz/Logical Operator

View

#:	Type	Description	Status	Test Case
1	Condition	Logic: input port 1 T	Satisfied	6
2	Condition	Logic: input port 1 F	Satisfied	1
3	Condition	Logic: input port 2 T	Satisfied	8
4	Condition	Logic: input port 2 F	Satisfied	6
5	Mcdc	Logic: MCDC expression for output with input port 1 T	Satisfied	8
6	Mcdc	Logic: MCDC expression for output with input port 1 F	Produced error	n/a
7	Mcdc	Logic: MCDC expression for output with input port 2 T	Satisfied	8
8	Mcdc	Logic: MCDC expression for		6

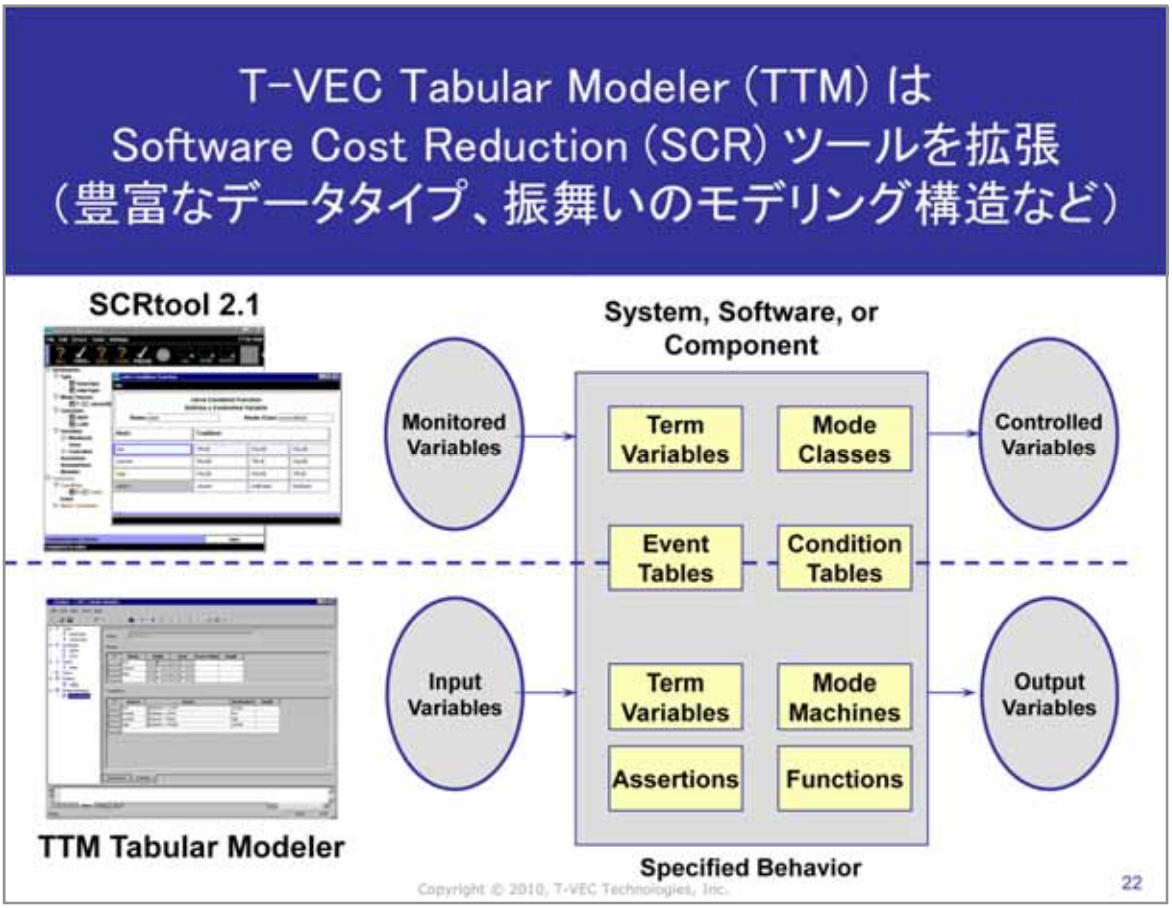


Test #	Vector #'s	_output	z_iPort	y_iPort
1	1, 3, 5, 13	1	-32768	-32768
2	2, 8, 10, 12	1	32767	32767
3	4	0	0	32767
4	6	0	32767	0
5	7	1	1	1
6	9	0	-1	32768
7	11	0	-32768	1
8	14	1	0	0

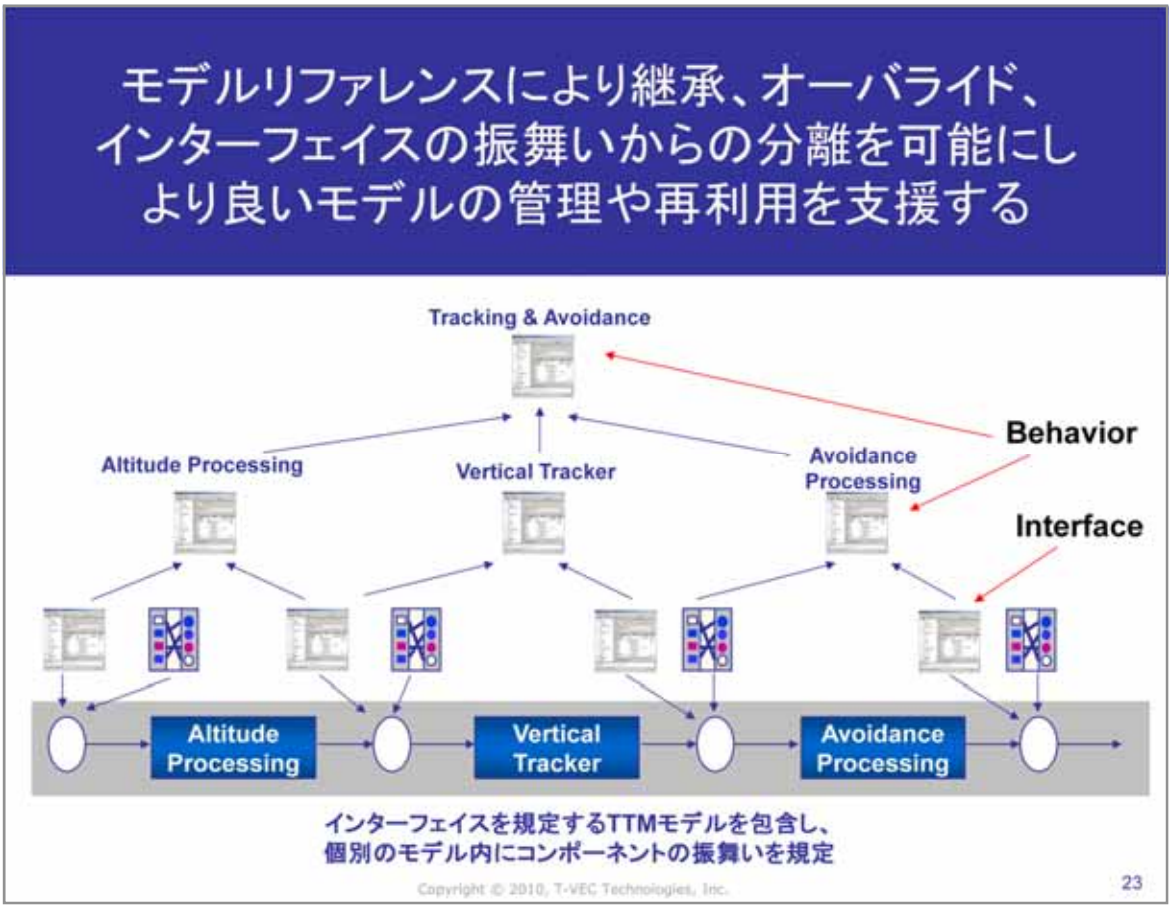
Validated by senior management from The Mathworks®

T-VEC VGS

21



1. TTM**はSCRツール･手法を拡張したもの**
2. SCR **は、**the Naval Research Lab back about 1980; also referred to as Parnas tables after Professor David Parnas **氏により開発された**
3. **オリジナルでは インテジャ、ブーリアン、フロートなど**(scalar data type)**のみであったが、構造体や配列もサポートできるようにした**
4. TTM added:
   - **要件管理**
   - **構造体、配列、ストリング**
   - **モデルリファレンス**
   - **要件へのトレーサビリティ**
   - **追加機能**(ln, log2, log, ceil, floor, round, truncate)
   - **パラメータ化された関数**
   - **アサーション**
   - **インライン化**



1. TTMへの拡張　モデルリファレンス、継承、オーバライドなど
2. モデルリファレンスによってインターフェイスを振舞いから分離し、一つのインターフェイスを定義して参照できるようになる
3. モデル上の変数と実装上のインターフェイスのためのオブジェクトマッピング
4. モデルの振舞いはインターフェイスのモデルを包含し、振舞いの組合せをサポートする
5. これによりモデルはシステムのアーキテクチャを正確に描写することになる
6. 統合テストをサポートするためのテストがモデルから生成できる

T-VEC Tabular Modeler (TTM) と テストベクタ生成システムの、 ドメインスペシフィックモデリングツールとの統合

Flight Control domain-Specific Language (FCSL)

TTM

T-VEC VGS

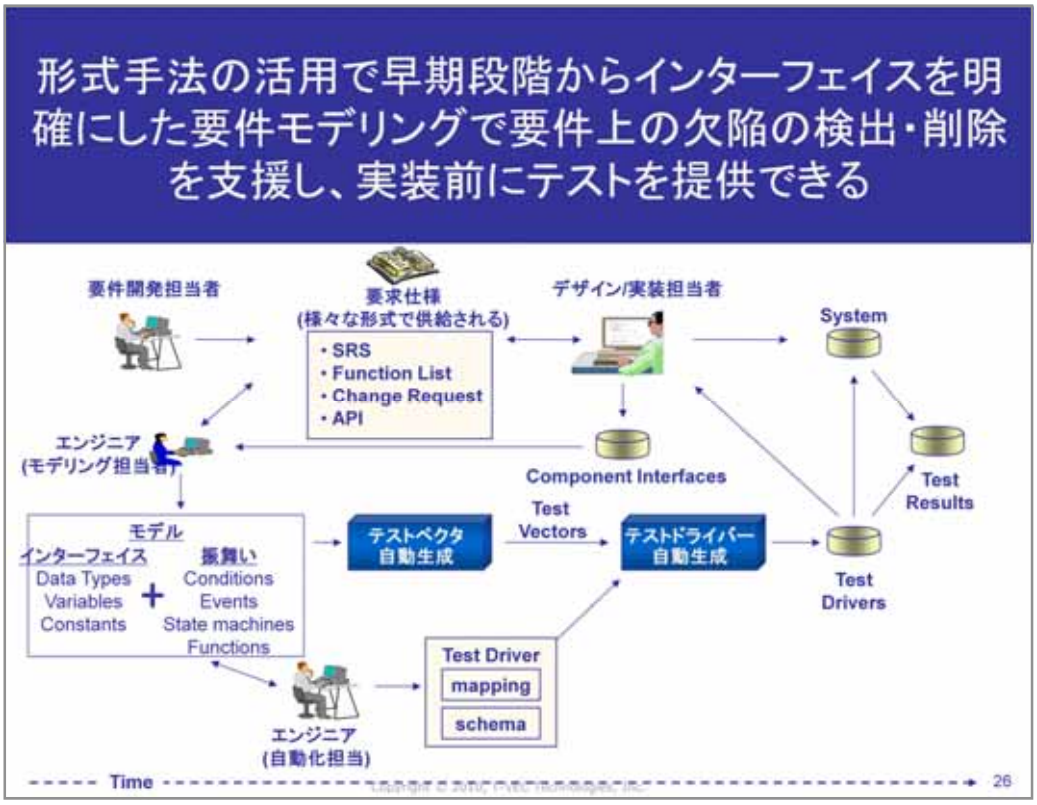
Producible Adaptive Model-based Software (PAMS) technology to the development of safety critical flight control software. PAMS has been developed under the Defense Advanced Research Projects Agency (DARPA) Disruptive Manufacturing Technologies program. Contract # N00178-07-C-2011.

MODEL-BASED ADAPTATION OF FLIGHT-CRITICAL SYSTEMS, *Sumit Ray, BAE Systems, Johnson City, New York, Gabor Karsai, Vanderbilt University, Nashville, Tenneessee , Kevin M. McNeill, BAE Systems, Arlington, Virginia, Digital Avionics Systems Conference, 2009*

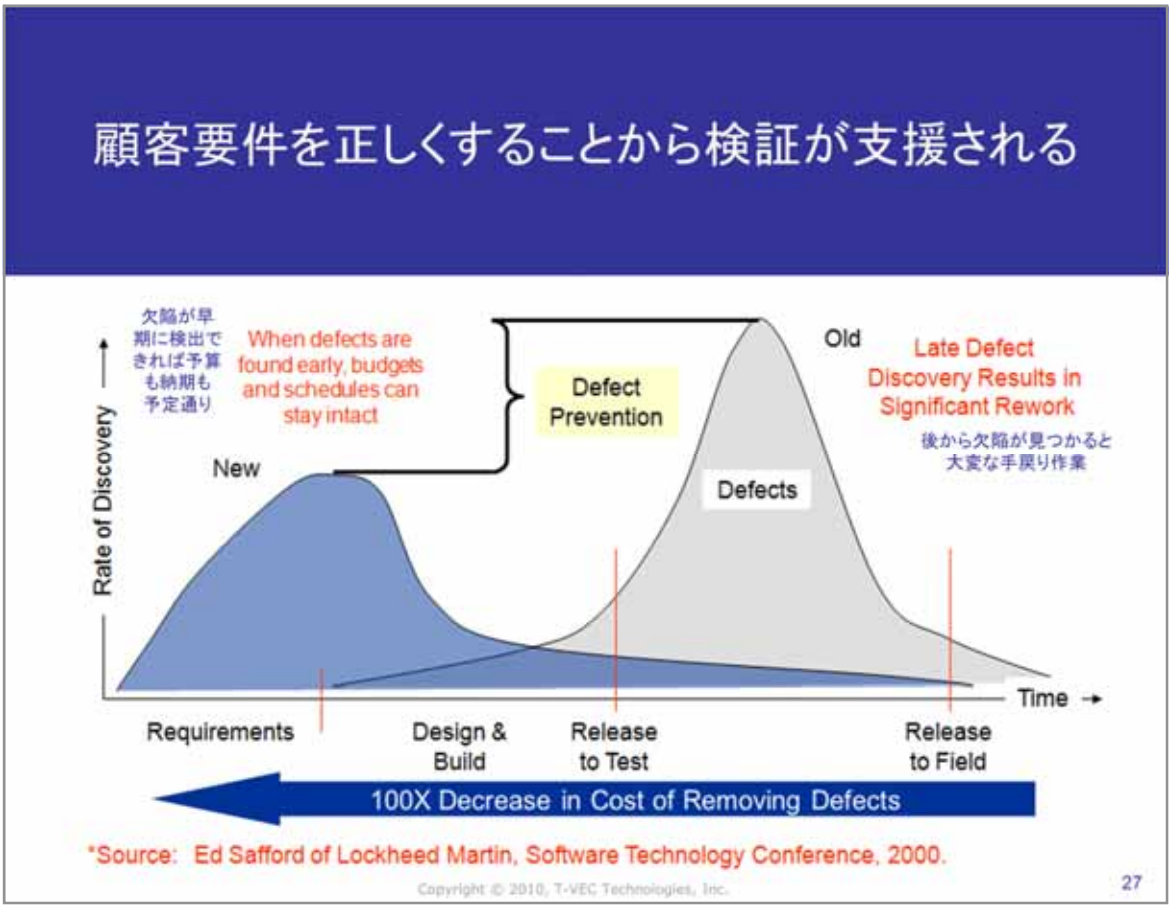
24

1. BAE**システム社、** *Vanderbilt University* **とドメインスペシフィックモデリング言語の活用について協力した**
2. TTM**モデル言語は**T-VEC**形式より抽象度が高い**
3. **それゆえドメインスペシフィックモデリング言語をモデル解析、テストベクタ生成に適した形式に変換することが容易**
4. SimulinkもDSML**であるが、大きくて複雑な言語 (ドメインが広いので)**

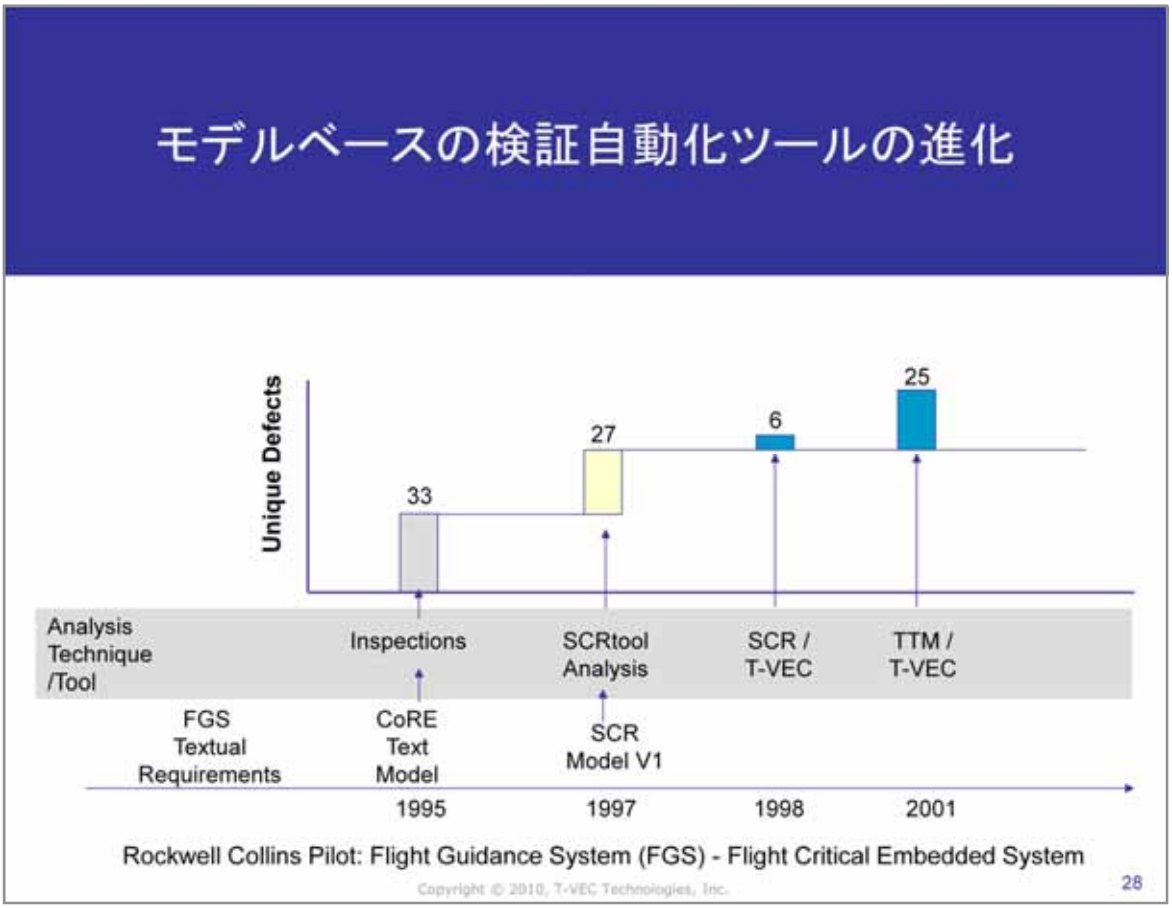
1. Lockheed Martin **社の許可を得た図**
2. **ここで肝心なことは開発費用の**70%**は早期段階で決定されてしまうということ**
3. **なぜソフトウエアシステムの検証費用が高価になるのか？** IC**開発者のように体系的で自動化される**V&V**を採用できないと、自らマニュアルで検証作業にあたることになり高価になってしまうということ。**
4. **複雑なセーフティクリティカルシステムでは**70%**以上にもなっている**



1. Lockheed Martin **社の実績**
2. **図の見方は左から右への時間経過**
3. **要件開発時からモデリングを採用し、要件上の問題は顧客と確認することなどを介して正すことができた。**
    - **いかに早期の要件検証が行えるかの実績**
4. **早期のインターフェイス解析を併合したことで、システムインターフェイスの観点から要件や派生要件をモデルに仕様化できる。**
5. **そして形式手法により定理証明、早期のテストベクタとテストドライバー生成を可能にし、デザイン担当者や実装担当者は早期段階から継続的にテストが行えるようになった**

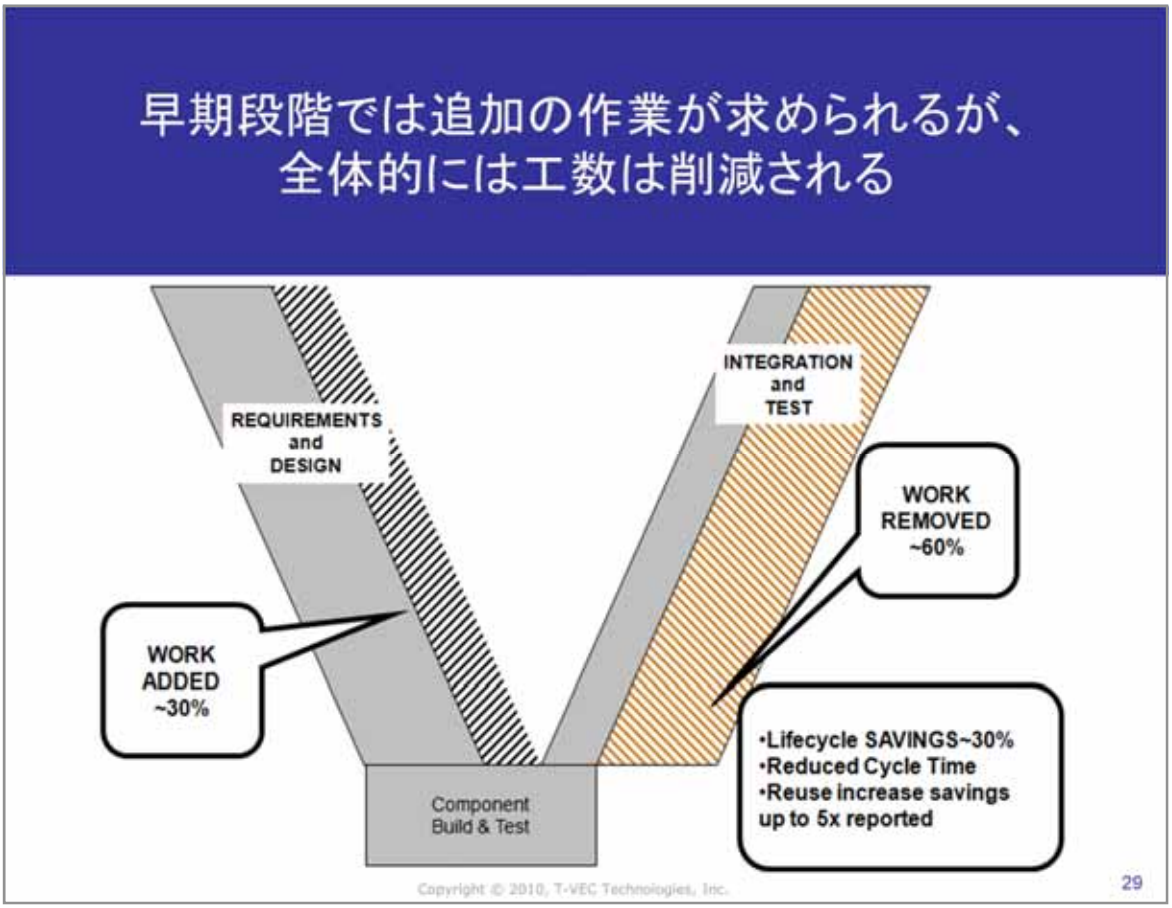


1. Ed Safford of Lockheed Martin at the STC conference just before the JSF award **からの図**
2. **モデリングにより欠陥を封じ込めることと、モデリングのプロセスで欠陥を未然に防ぐことを証明した**
3. **多くの欠陥は形式手法による定理証明によって早期に認識され、手戻り作業を削減できた**



1. Rockwell Collins on a Flight Guidance Mode System **社での**Fagan inspection **の実績**
2. **当初はテキストドキュメントを**CoRE**ツール**（**テキストベースの**SCR**へ**）
3. SCR**ツールがリリースされ追加の欠陥を検出し、それが進化して**T-VEC**の形式手法**（**定理証明＋テストベクタ生成**）**と統合されることで更なる欠陥を検出できるようになった**

> *Fagan inspection* refers to a structured process of trying to find defects in development documents such as programming code, specifications, designs and others during various phases of the software development process. ... http://en.wikipedia.org/wiki/Fagan_inspection



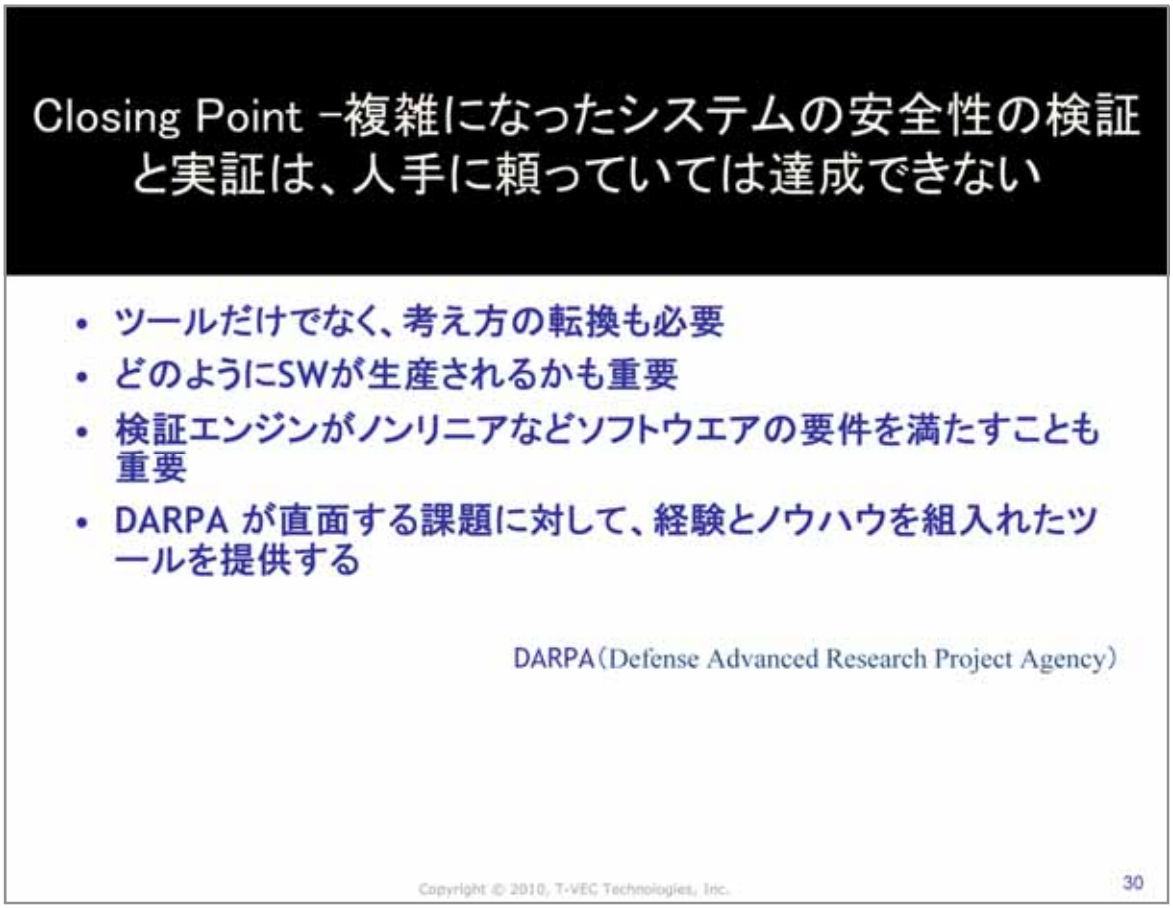
1. Lockheed Martin, Rockwell, NIST などによりこれらの実践的な事例が公開されているが、多くの企業は公開したがらない (C-5の事例のように既存製品との比較が出ることが躊躇されるので)

2. この図は上述企業とは別の会社で、開発ライフサイクルへの影響を考察するために作成された

- モデリングにより上流工程に追加の工数が求められるが、下流工程を削減した
- モデルの再利用により5倍の効率になることは重要なポイント (既存モデルを再利用して拡張することができる)
- モデルの機能は製品ごとで共通なので、運用上の再利用 (共通ツールに対する理解・経験) なども効率化できる

Organizational Best Practices

インターフェイスドリブンなモデリングは開発しながら適応させることに飛躍的な効果が認められた。理想的にはテストエンジニア (モデラー) が開発者と共にインターフェイスを安定させて、要件を精練し、イテレーティブに反復して開発・テストできるようなモデルを作る。テストエンジニアはモデルを介して製品の要件 (通常殆ど不完全なままの) を精練する、それは数百・数千行ものテストスクリプトを書く代わりに。そして形式手法を用いて仕様・設計モデルを定理証明し、テストベクタ (入力と期待値) ・テストドライバーを自動生成。イテレーティブに反復される開発の中、コンポーネントの振舞い、インターフェイス、要件などが変更される場合、モデルは修正され、テストケースとドライバーが再生成され、再実行できる。このツールの特筆すべき優位性は、テストのプロセスが通常の開発と同時に行えること。Lockheed Martin 社など主な顧客は、50%以上テストの工数が削減できたことや、早期段階で要求仕様の解析 (形式手法による定理証明) をすることで要件上の欠陥を排除し手戻り作業を飛躍的に軽減できたことを公言している



1. **先見の明、洞察力を持った推進派が必須だが、、、**
2. **試みは底辺からスマートなエンジニア達によって取組まれることが多いが、実プロジェクトに適合させ調整することを学ぶ時間や予算が十分ではない**
   - **どのようにMDEツールを使うかを決定することが先決（いきなり取り入れない）**
   - **講習でツールの動作は覚えても、ツールでどのように開発していくかがわからない**
3. **検証が容易になるようなモデリング**

Terms and Acronyms

AADL Architecture Analysis & Design Language
AP233 Application Protocol 233
ATL ATLAS Transformation Language
BPML Business Process Modeling Language
CAD Computer-Aided Design
CASE Computer-Aided Software Engineering
CATIA Computer Aided Three-dimensional Interactive Application
CDR Critical Design Review
CMM Capability Maturity Model
CMMI Capability Maturity Model Integration
CWM Common Warehouse Metamodel
DBMS Database Management System
DoDAF Depart of Defense Architectural Framework
DSL Domain Specific Languages
EMF Eclipse Modeling Framework
GME Generic Modeling Environment
IBM International Business Machines
ICD Interface Control Document
IEEE Institute of Electrical and Electronics Engineers
INCOSE International Council on Systems Engineering
IPR Integration Problem Report
ISO International Organization for Standardization
IT Information Technology
JET Java Emitter Template
Linux An operating system created by Linus Torvalds
MAP Modeling Adoption Practices
MARTE Modeling and Analysis of Real Time Embedded systems
MATRIXx Product family for model-based control system design produced by National Instruments
MBT Model Based Testing
MDA® Model Driven Architecture®
MDD™ Model Driven Development
MDE Model Driven Engineering
MDSD Model Driven Software Development
MDSE Model Driven Software Engineering
MIC Model Integrated Computing
MMM Modeling Maturity Model
MoDAF United Kingdom Ministry of Defence Architectural Framework
MOF Meta Object Facility
MVS Multiple Virtual Storage
NASA National Aeronautics and Space Administration
OCL Object Constraint Language
OMG Object Management Group
OO Object oriented
PDR Preliminary Design Review
PIM Platform Independent Model
Pro/E Pro/ENGINEER
PSM Platform Specific Model
QVT Query/View/Transformation
RFP Request for Proposal
ROI Return On Investment
RTW Mathworks Real Time Workshop
SSCI Systems and Software Consortium
Simulink/Stateflow Product family for model-based control system produced by The Mathworks
SCR Software Cost Reduction
SDD Software Design Document
SOAP A protocol for exchanging XML-based messages - originally stood for Simple Object Access Protocol
Software Factory Term used by Microsoft
SQL Structured Query Language
SRS Software Requirement Specification
SysML System Modeling Language
SystemC IEEE Standard 1666
UML Unified Modeling Language
XMI XML Metadata Interchange
XML eXtensible Markup Language
xUML Executable UML
Unix An operating system with trademark held by Open Group
VHDL Verilog Hardware Description Language
VGS T-VEC Vector Generation System
VxWorks Operating system owned by WindRiver

Trademarks

- OMG®, MDA®, UML®, MOF®, XMI®, SysML™, BPML™ are registered trademarks or trademarks of the Object Management Group.
- IBM™ is a trademark of the IBM Corporation
- Java™ and J2EE™ are trademark of SUN Microsystems
- XML™ is a trademark of W3C
- BridgePoint is a registered trademark of Mentor Graphics.
- Java is trademarked by Sun Microsystems, Inc.
- MATRIXx is a registered trademark of National Instruments.
- Real-time Studio Professional is a registered trademark of ARTiSAN Software Tools, Inc.
- Rhapsody is a registered trademark of Telelogic/IBM.
- Rose XDE is a registered trademark of IBM.
- SCADE is copyrighted to Esterel Technologies.
- Simulink is a registered trademark of The MathWorks.
- Stateflow is a registered trademark of The MathWorks.
- Statemate is a registered trademark of Telelogic/IBM.
- T-VEC is a registered trademark of T-VEC Technologies, Inc.
- UNIX is a registered trademark of The Open Group.
- VxWorks is a registered trademark of Wind River Systems, Inc.
- VectorCAST is a trademark of Vector Software.
- Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
- All other trademarks belong to their respective organizations.