Model-Driven Verification and Validation

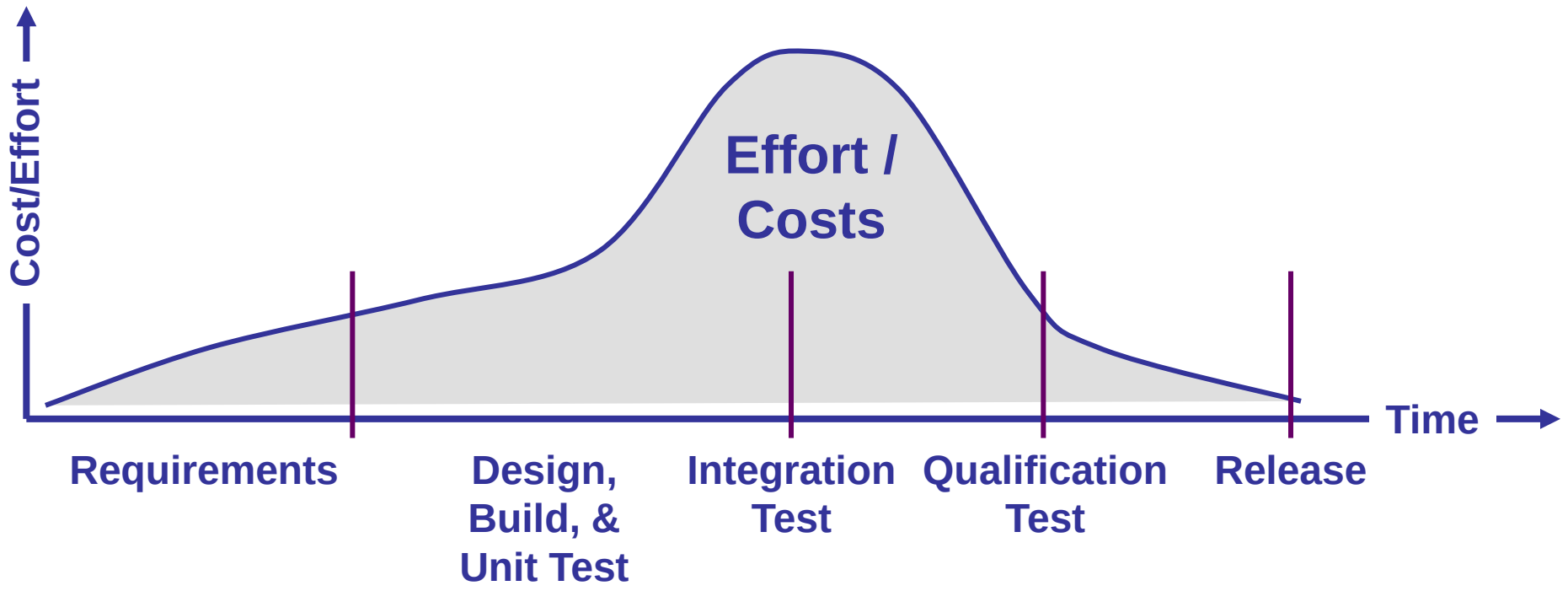
Presented at:
Safe & Secure Systems & Software Symposium
June, 2010

By
Mark R. Blackburn, Ph.D.

SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

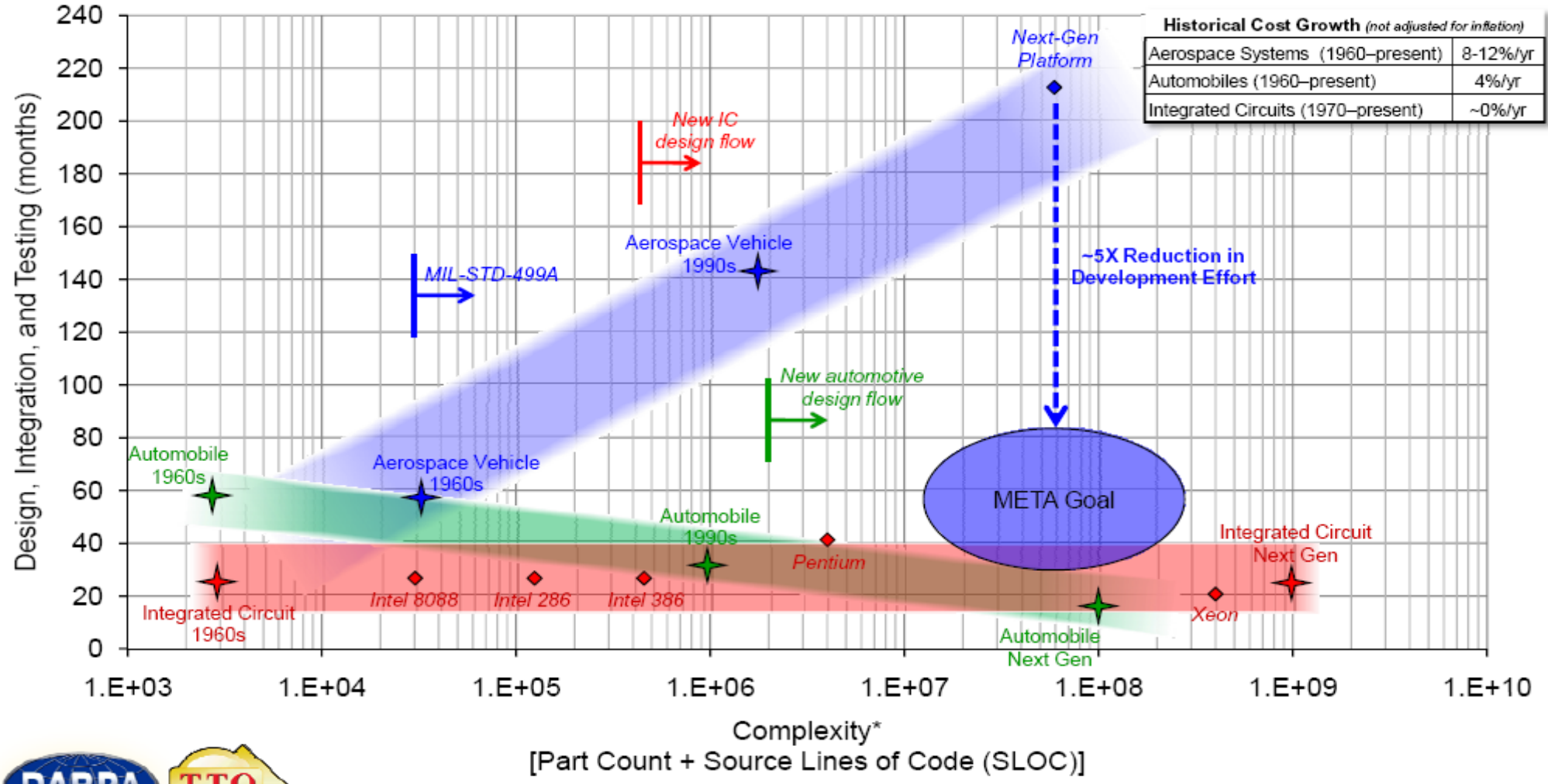
T-VEC Technologies, Inc.

Point A: 1985 - Verification for safety-critical system was 70% of total cost; many people added to project to test (mostly manual)



2009 - META Program: Failure to update 1960s- systems engineering process significantly increases cost and schedule

Point B: We must produce systems of the same complexity as hardware with similar costs and schedules



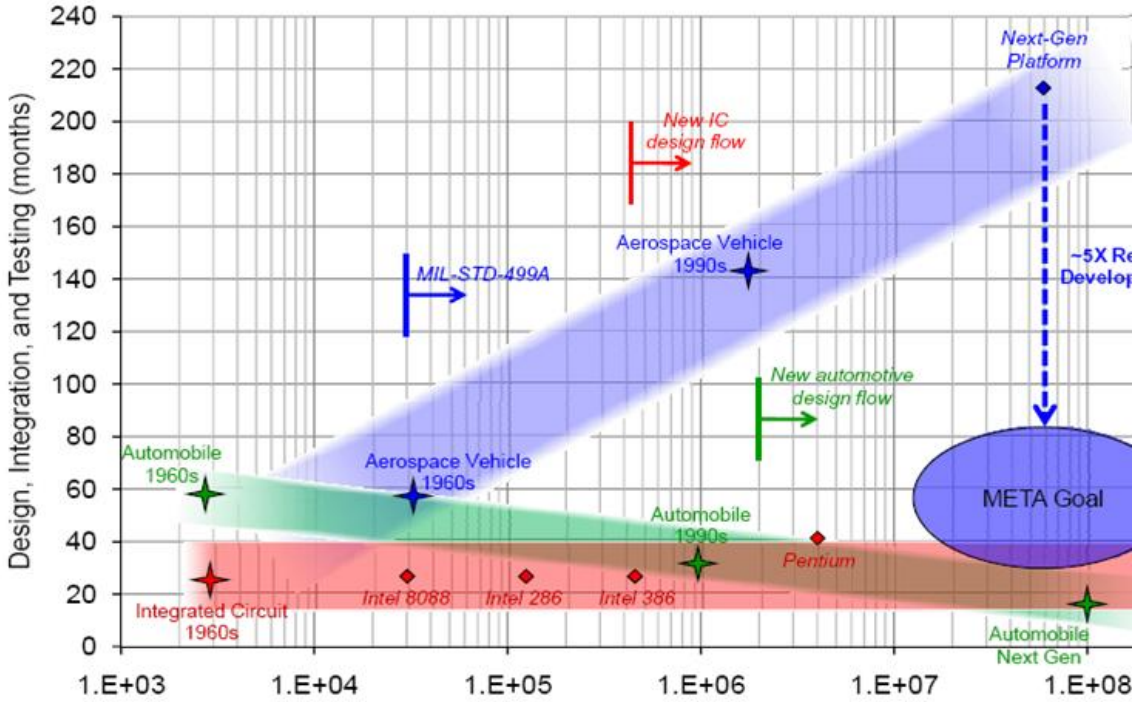
Note (*): Not a great metric. But that's what we have today. META will come up with better metrics.

Joseph Sifakis, one of the 2007 ACM A.M. Turing Award winners, discussed things about modeling and verifying software systems

- **First of all, it can be very challenging to construct faithful mathematical models of complex systems.**
- **For hardware, it's relatively easy to extract mathematical models, and we've made a lot of progress.**
- **For software, the problem is quite a bit more difficult. It depends on how the software is written, but we can verify a lot of complex software.**
- **But for systems consisting of software running on hardware, we don't know how to construct faithful mathematical models for their verification.**

* Quotes from Talking Model-Checking Technology by Leah Hoffman
(A conversation with Joseph Sifakis the 2007 ACM A.M. Turing Award winners.)

Call to action:
We have to address the differences between software-systems and hardware (ICs)



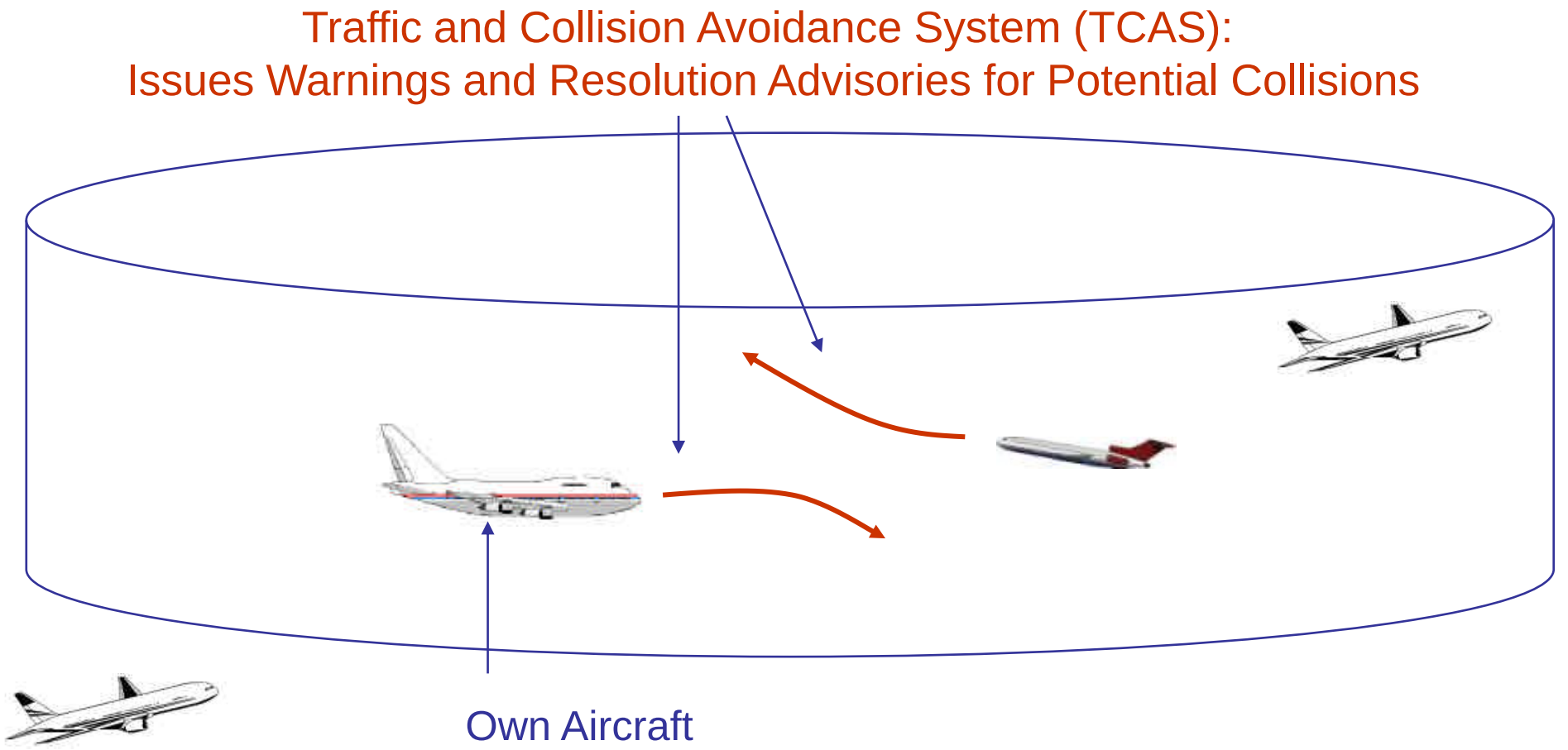
What's Different?

Software behavior often relies on floating point variables with non-linear relationships and constraints

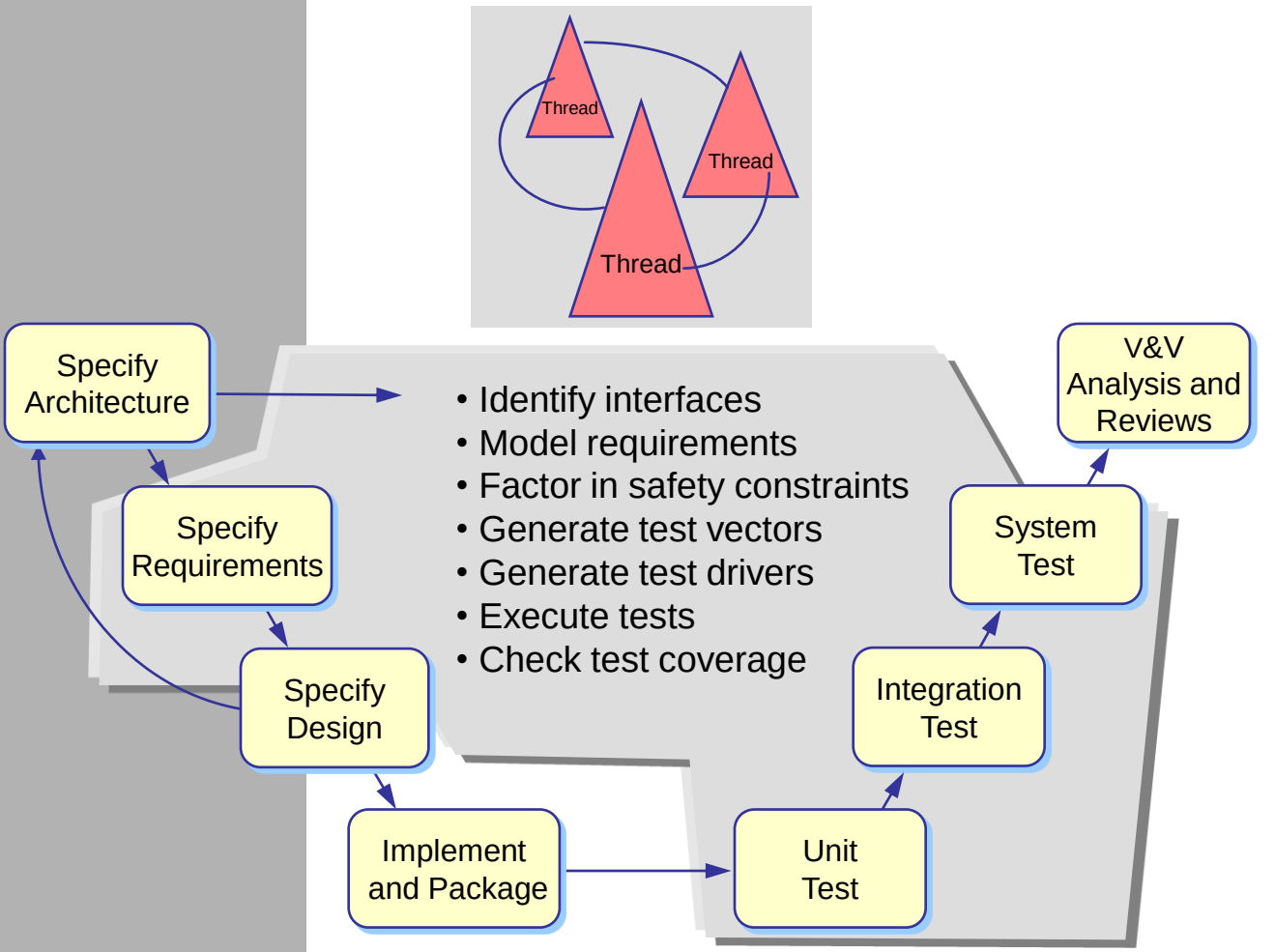
Three things matter

- How you write the software matters
  - Models have to map to the implementation
- "Verification engine" needs to be powerful
  - Need to cover every type of modeling construct, even non-linear ones
- Modeling has to be "easy" to use
  - Don't need to understand theorem proving

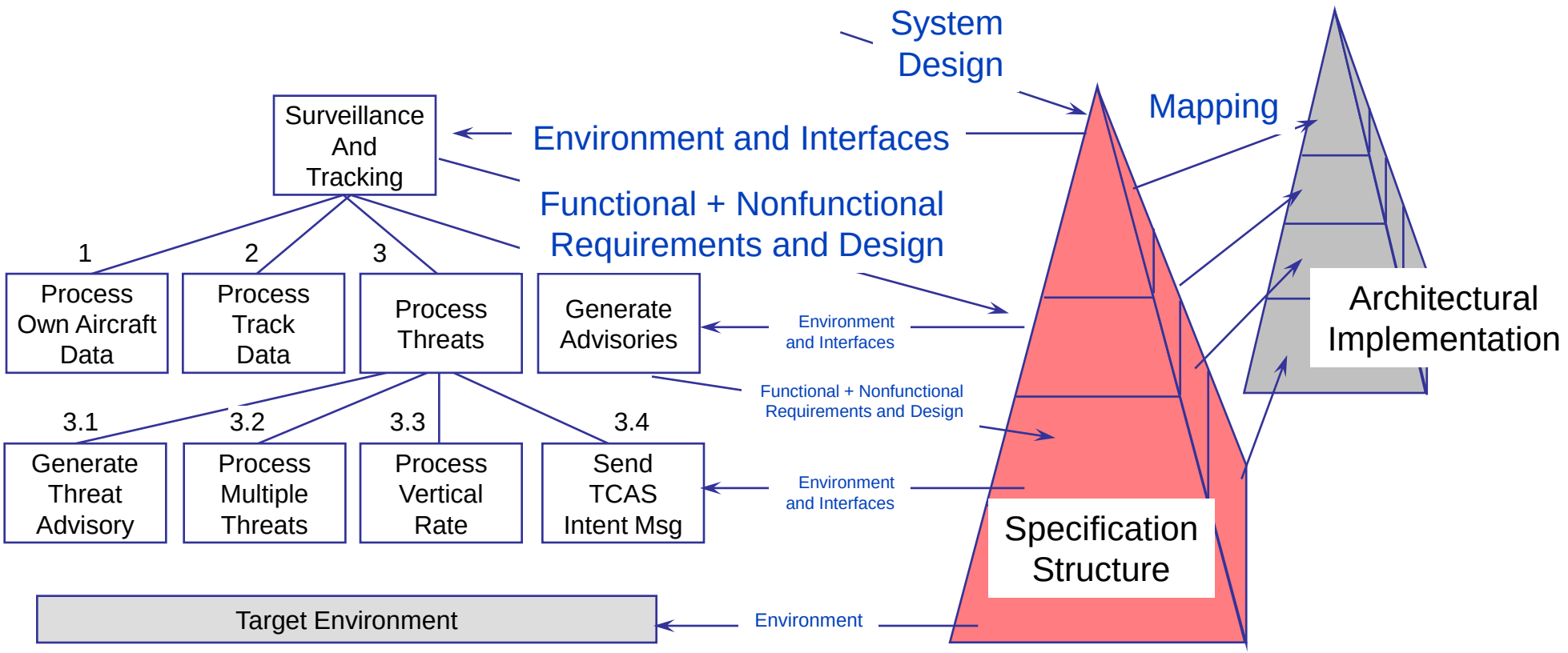
Key Point #1 – 1988 - we started with the problem where we had to address model-based verification of non-linear functions and constraints for the surveillance and tracking of aircraft



Used constructive V&V approach based on recursive modeling that mapped to implementation to produce V&V evidence during development



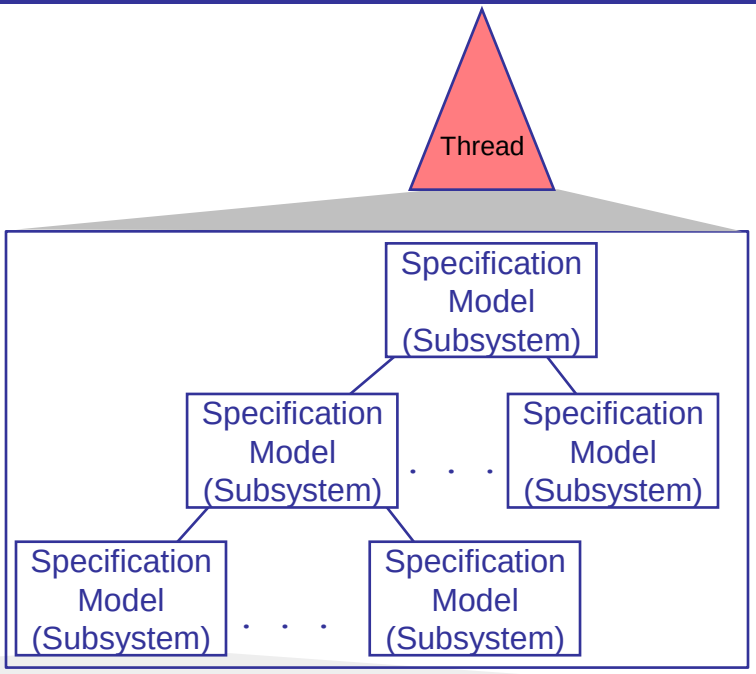
Hierarchical specification (model) addressing implementation-derived requirements while ensuring design for controllability and observability to support unit, integration and system testing

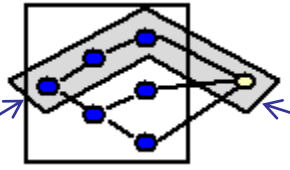


Similar to Simulink Models

Threads are hierarchies of subsystems transformed at the lowest level into preconditions and postcondition sets

- **A precondition (DCP) defines ANDed set of constraints on inputs**
- **A postcondition defines the output as a function of the constrained inputs**





DCP: Domain Convergence Path

Precondition — defines

Postcondition — used as inputs

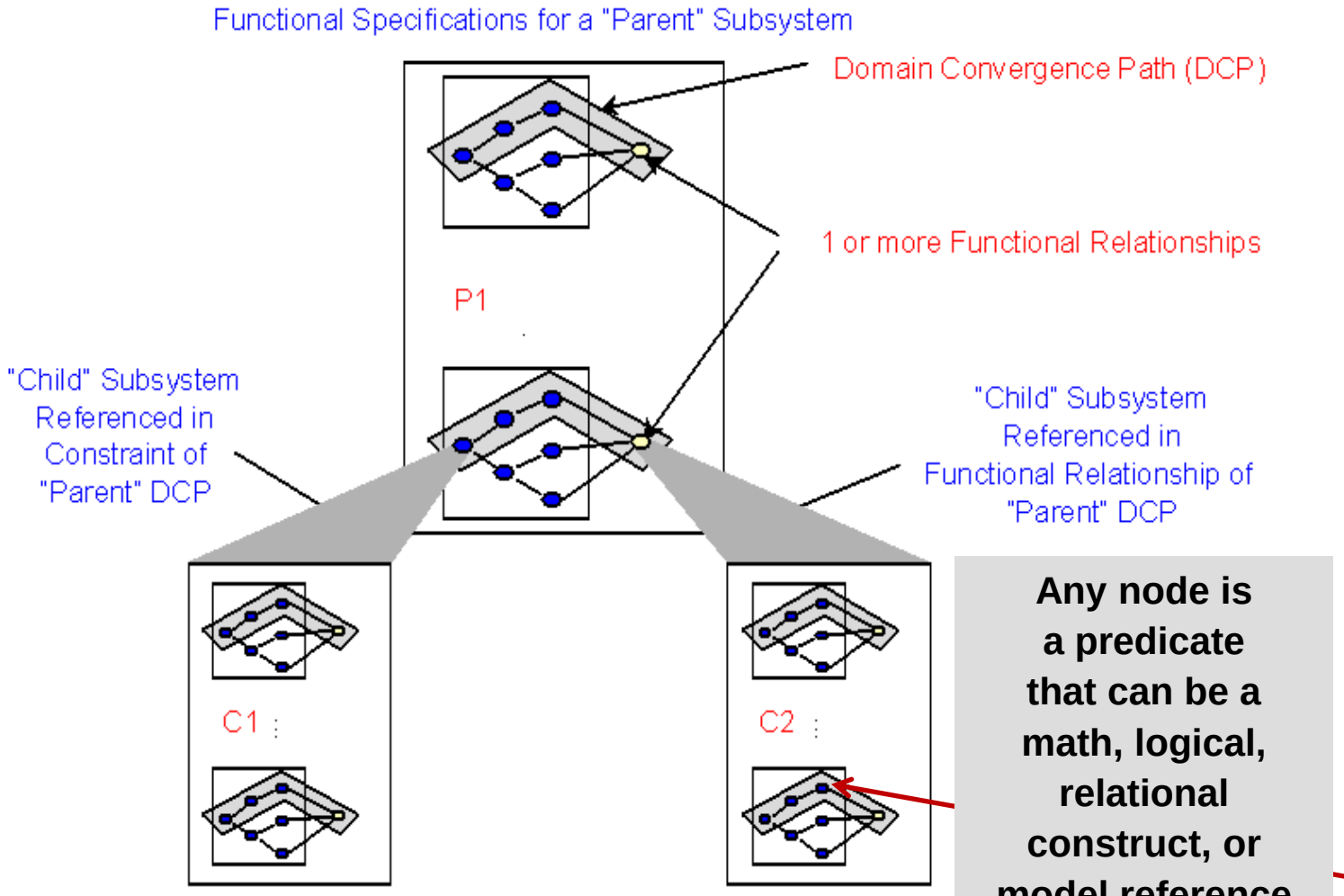
DCP: Constraint 1 ⟶ output = f_1(inputs)

OR

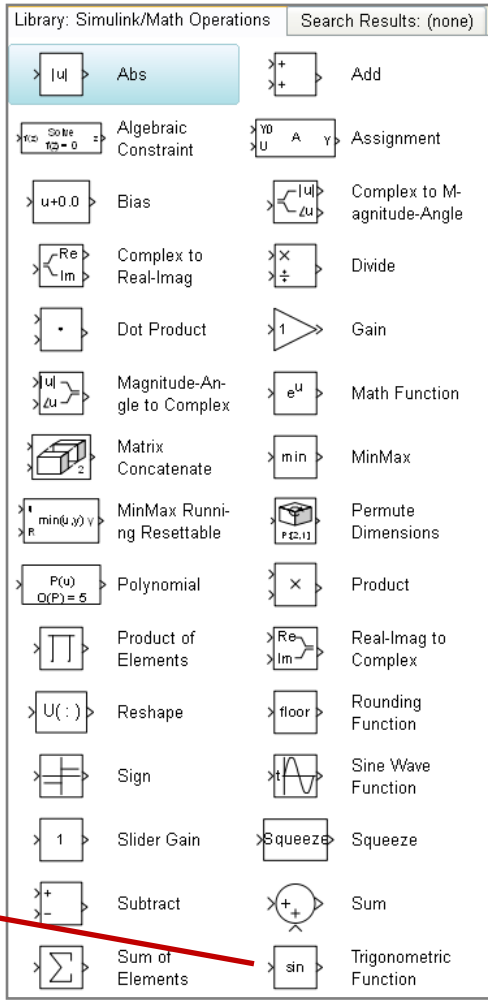
DCP: Constraint 2 ⟶ output = f_2(inputs)

. . .

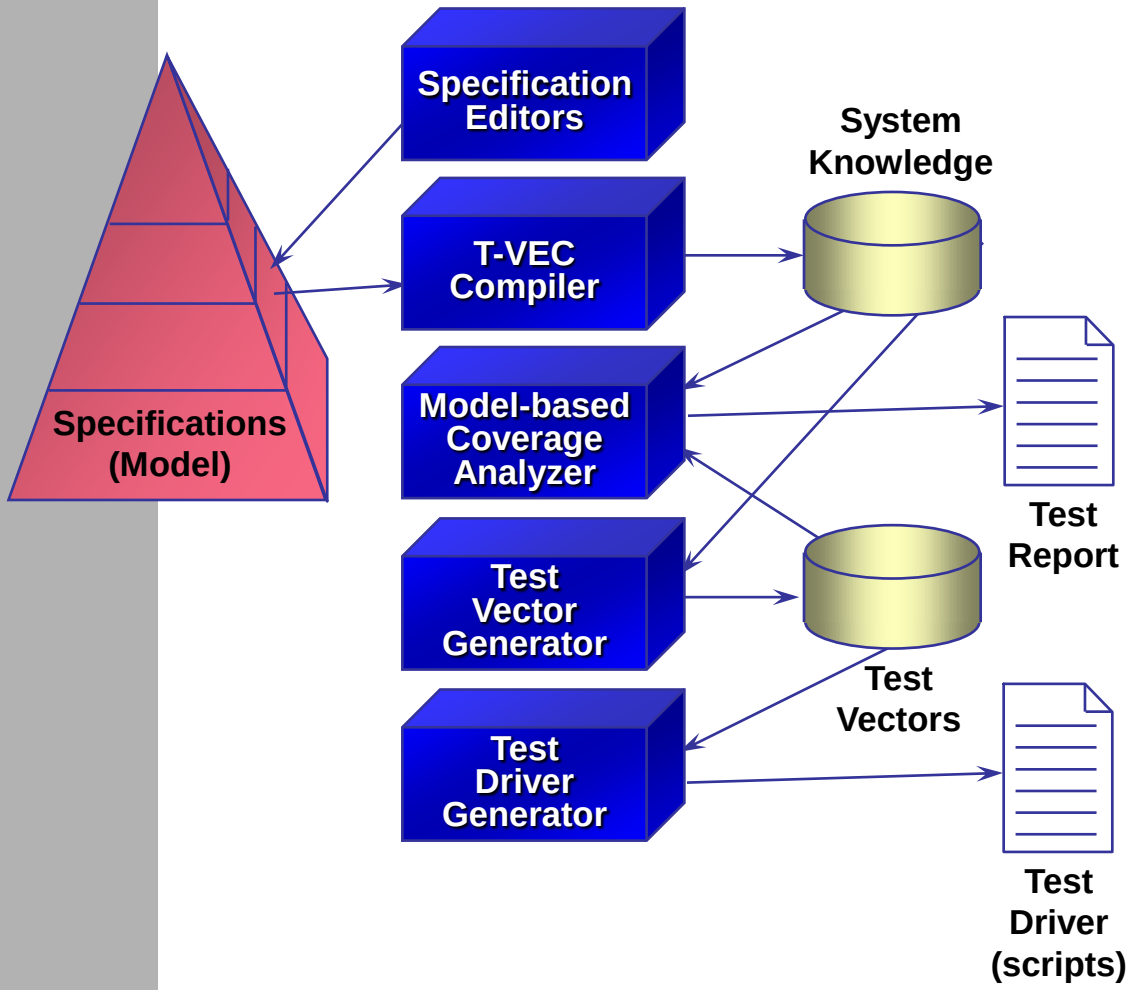
Test vectors are generated for each domain convergence path for all hierarchical subsystems if the constraints are satisfiable



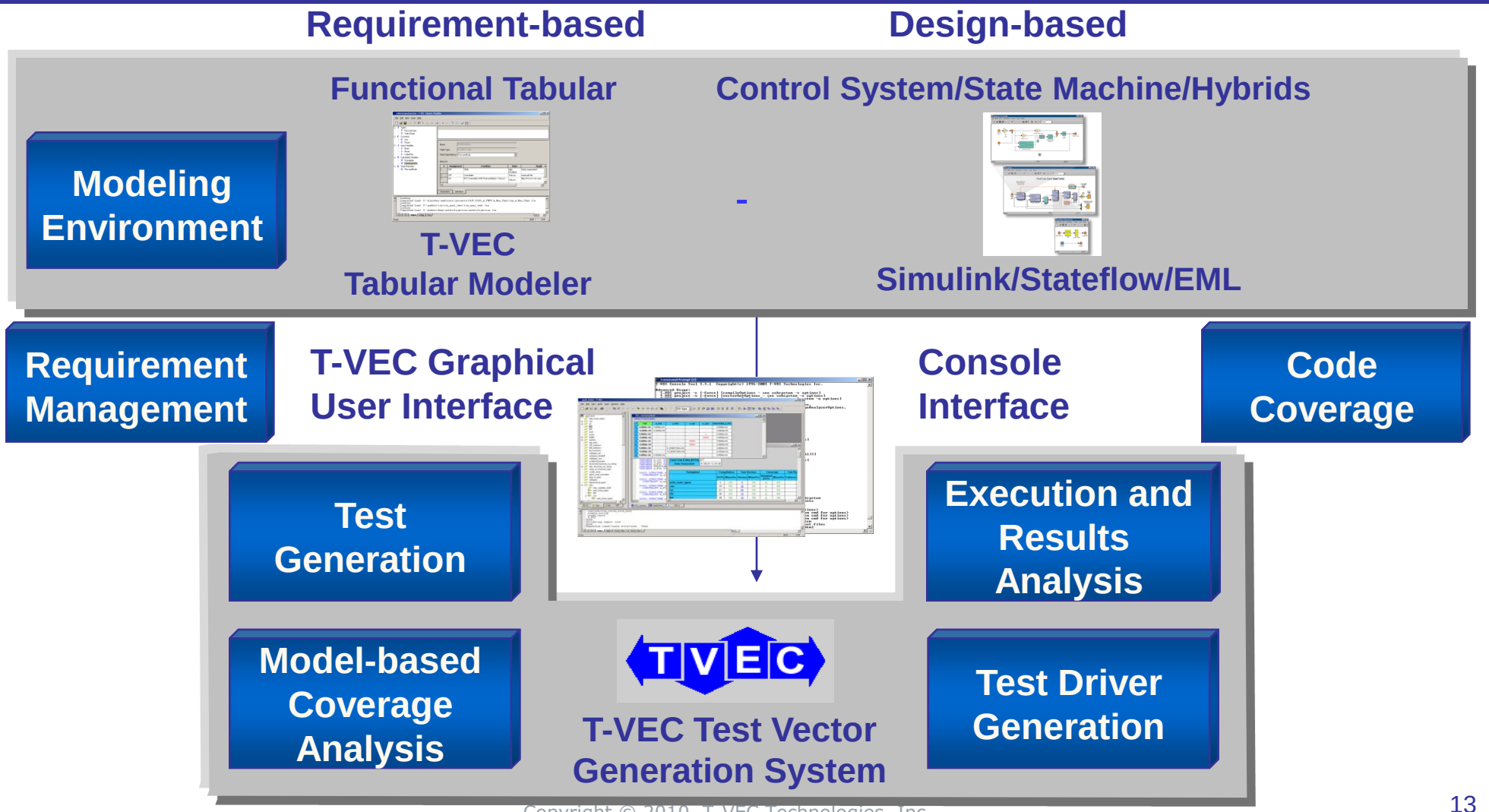
For example



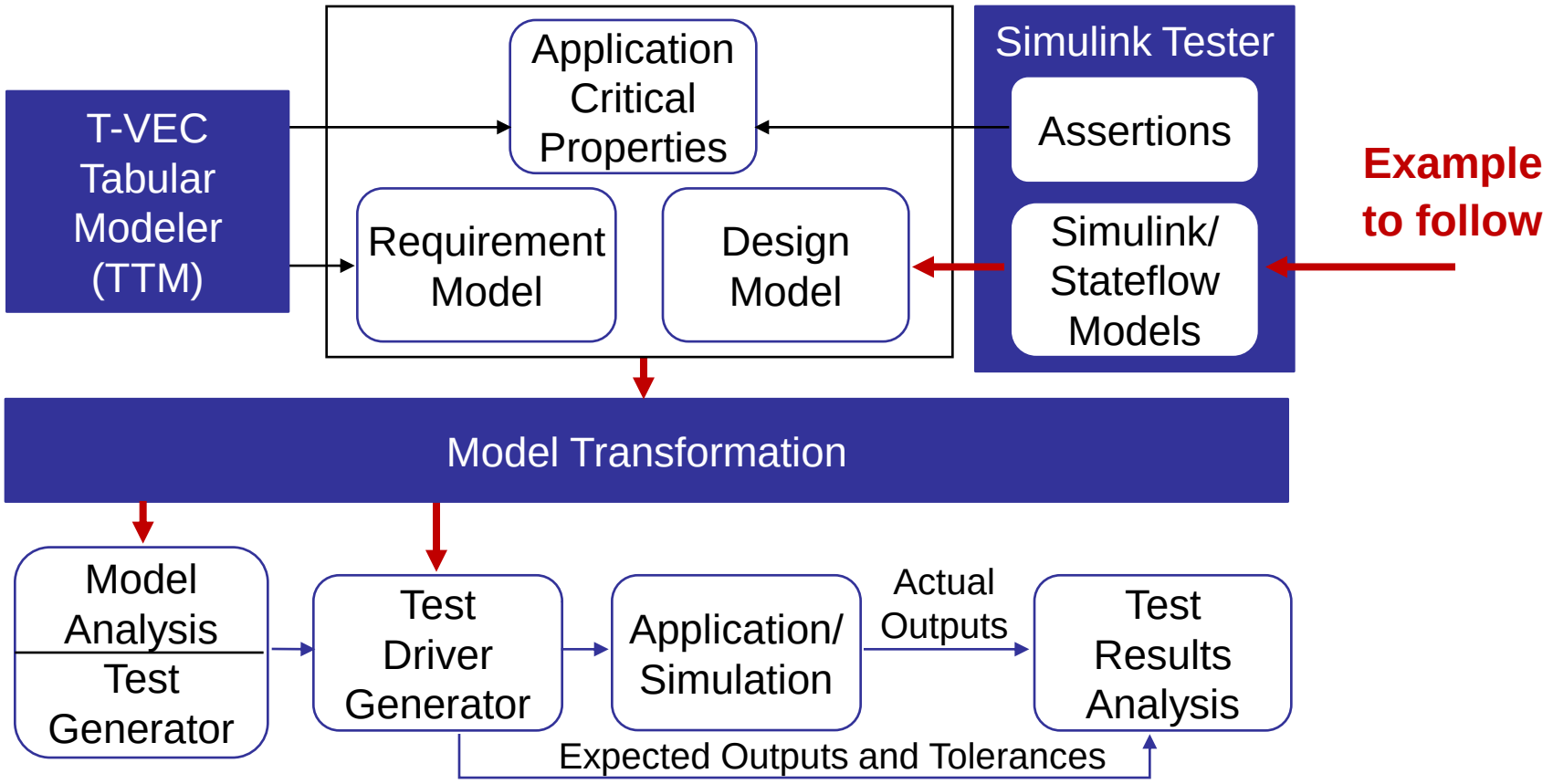
1988 – fundamental concepts of first generation tools still relevant today



1995 created a framework for integrating modeling and other related tools focused on model analysis and test automation with requirement-to-test traceability

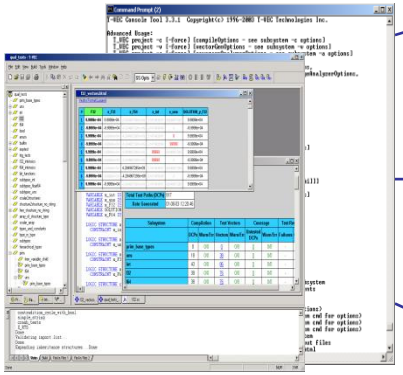


Transformed models are analyzed by theorem prover to ensure precondition is satisfiable; test inputs selected at subdomain boundaries and expected outputs generated for testing application



DCP: Domain Convergence Path

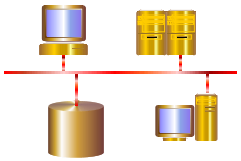
Test driver generation uses generic test vectors, object mappings and test driver schema to produce a driver that can run on host, target, or simulation environments



Test Driver Generation

Test Driver Languages

- Java
- C++
- Ada
- Perl
- SQL/ODBC/JDBC
- XML
- SOAP
- WinRunner
- JCL
- Python
- Basis and VB
- Custom (graphics)
- Assembler
- shell
- command languages
- emulators
- proprietary
- more . . .

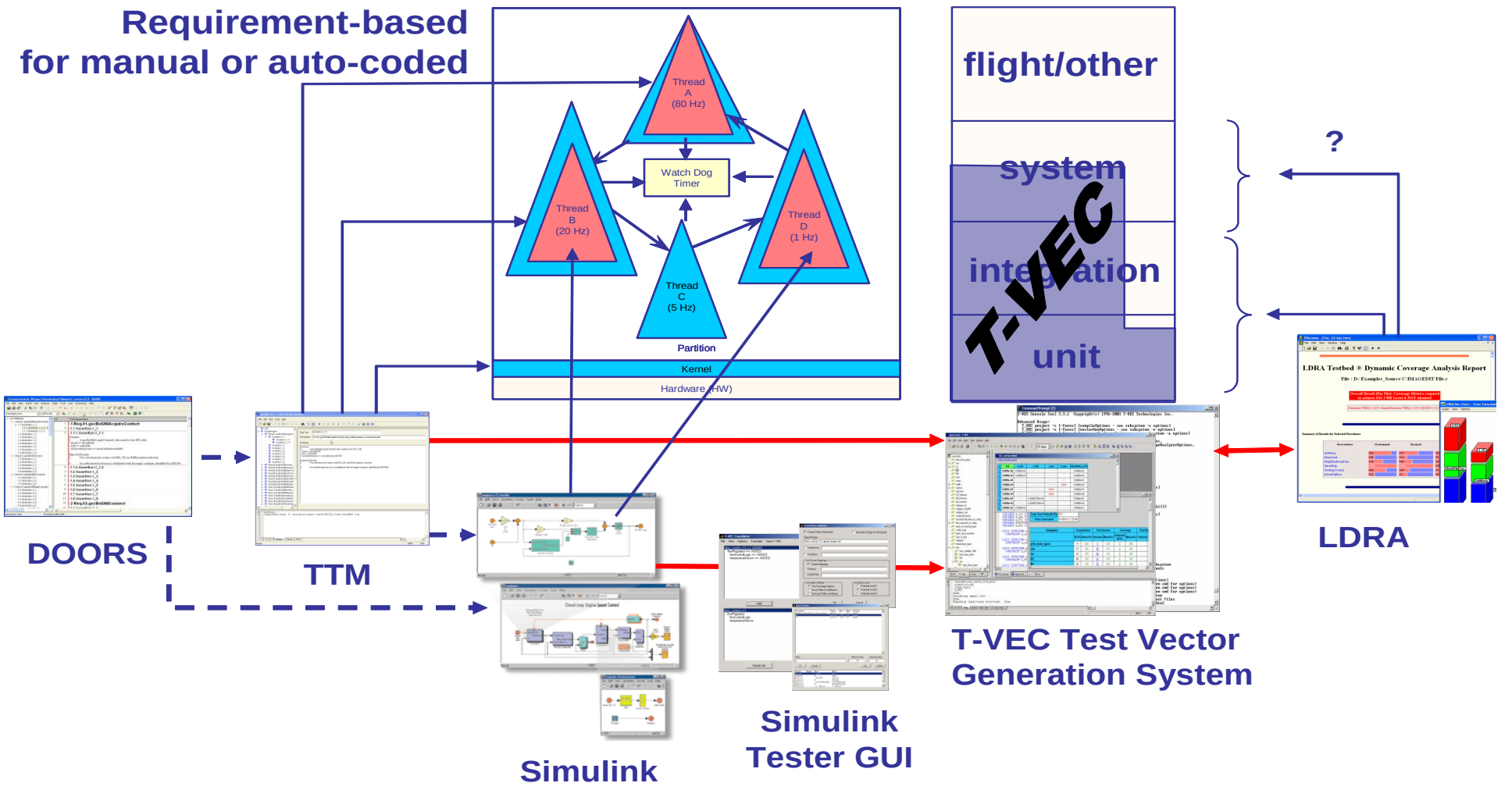


Database system
Client server
Web-based systems

Mission/life critical systems and high dependability components

Software modules for unit and integration testing

Key Point #2 – tool chains are emerging, because no one tool solves the entire problem, and we need to leverage the distinguishing capabilities of tools



Analysis capabilities often needed to ensure model is defect free before code generation and verification

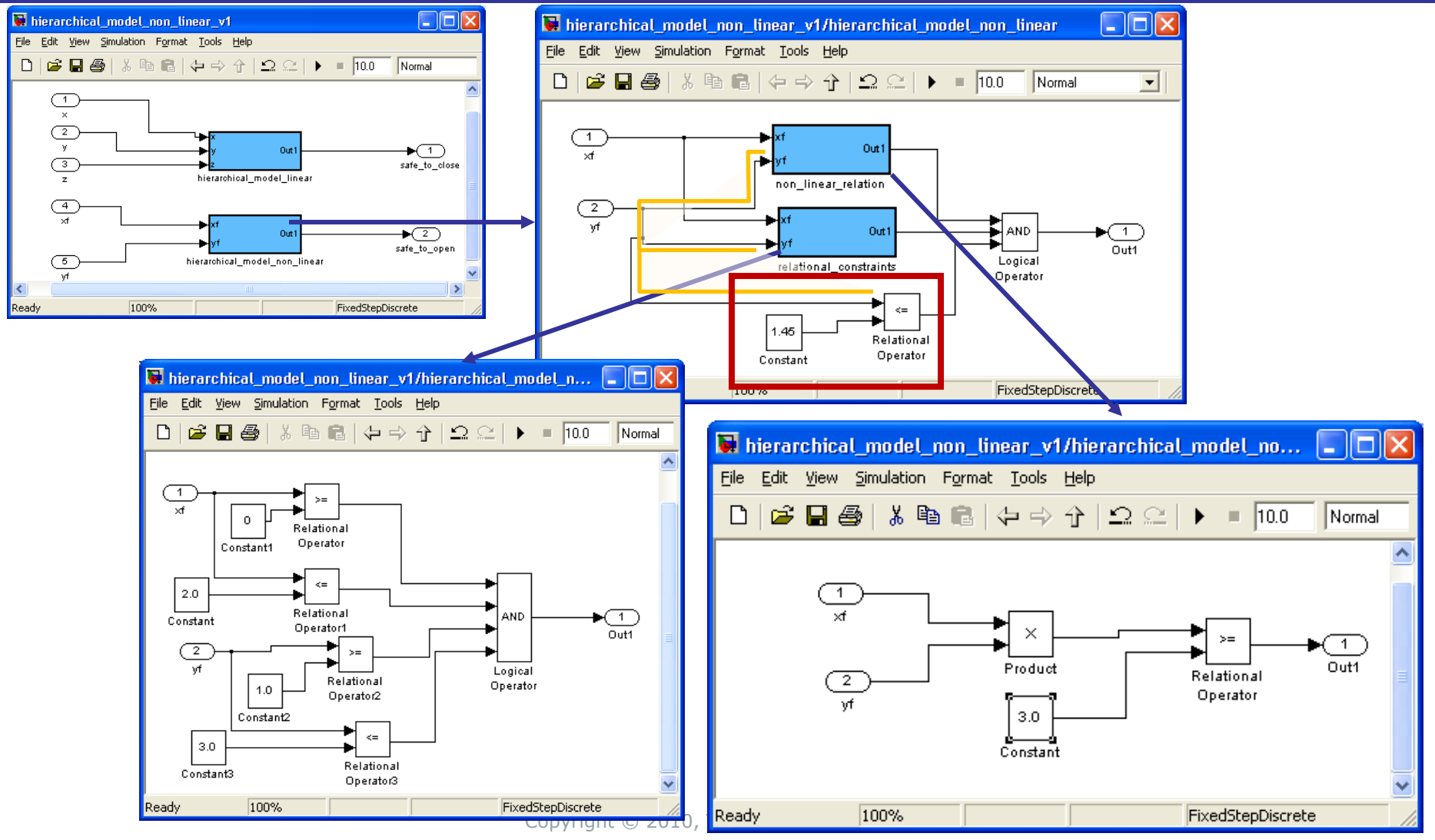
(Tool capabilities vary significantly)

	MS Project	DOORS	TTM/SCR	Simulink	Stateflow	Real Time Workshop	T-VEC Tester for Simulink	T-VEC Vector Generation System	Design Verifier with Prover	MS System Test	Simulink V&V	Vectorcast
Project planning	H											
Requirement management		H	S/I									
Requirement modeling			H									
Requirement simulation			I					S				
Design modeling				H								
Design simulation				H								
Concurrent modeling												
Property specification			S				H					
Model transformation			H				H	I				
Code generation				I	I	H						
Disjointness checking			I					H				
Race condition checking			I					H				
Satisfiability non-linear								H				
Satisfiability linear								H	L			
Satisfiability logical								H	S			
Arithmetic exceptions								H				
Prove properties			I	I	I			H	S			
Concurrency analysis				I	I	S						
Test sequence generation							I	H				
Test vector generation								H				
Test driver generation								H				I
Model coverage								H	I		H	
Test coverage analysis												H
Test execution												H
Test results analysis								H				H
Model-to-test traceability			I	I				H				
Model standards											H	
Code standards												H
Model management												
Confirmation management												
Test management								A		H		H
Embedded Target Testing												H

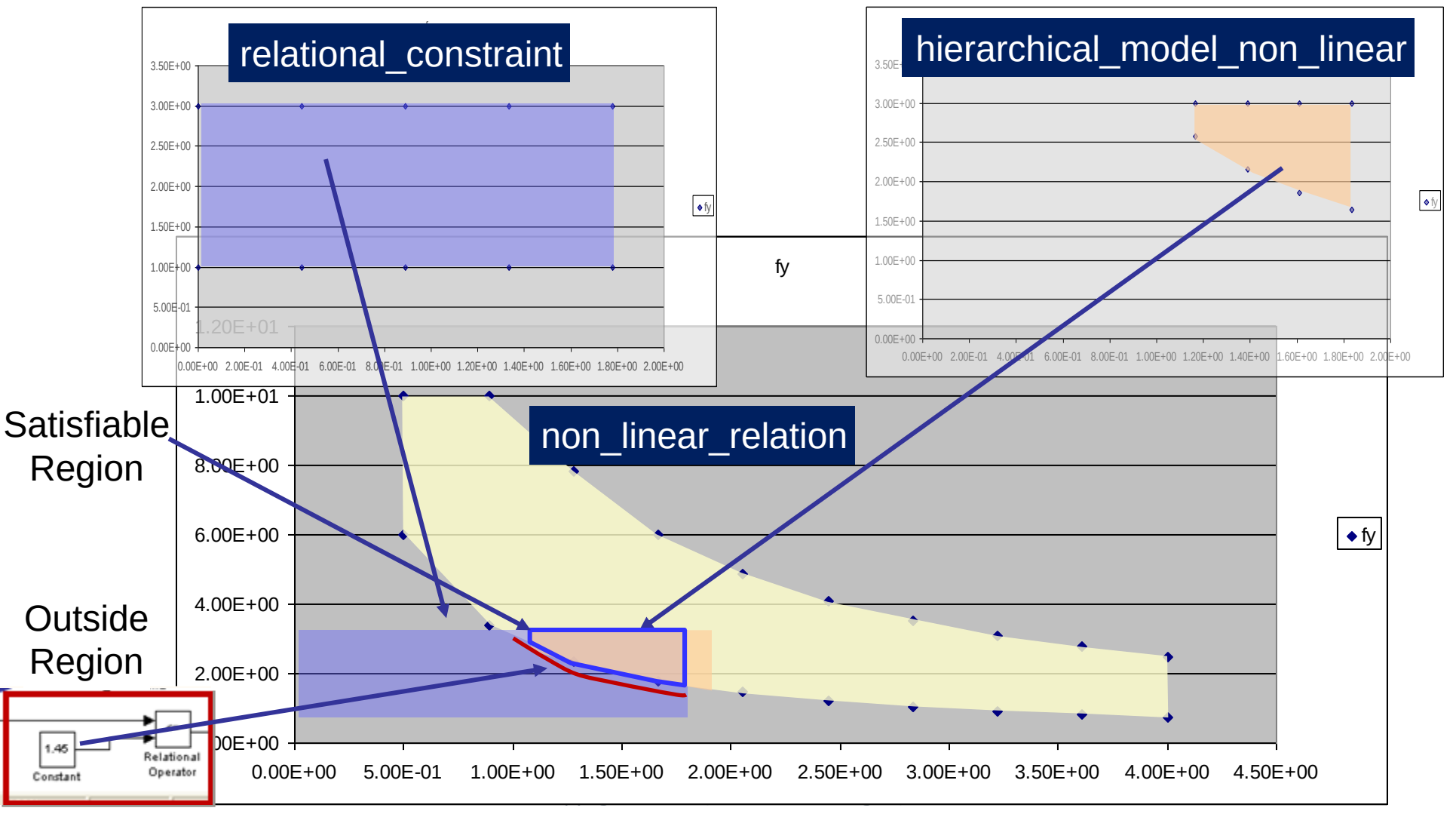
Key: Categories

- H: High - relatively complete
- S: Some
- L: Low
- U: Unknown - possible
- A: Absorbed - unnecessary
- I: Provided through tight tool integration

Example Simulink model seeded with a defect involving trivial non-linear operation involving floating point signal with goal to see if tools could identify defect



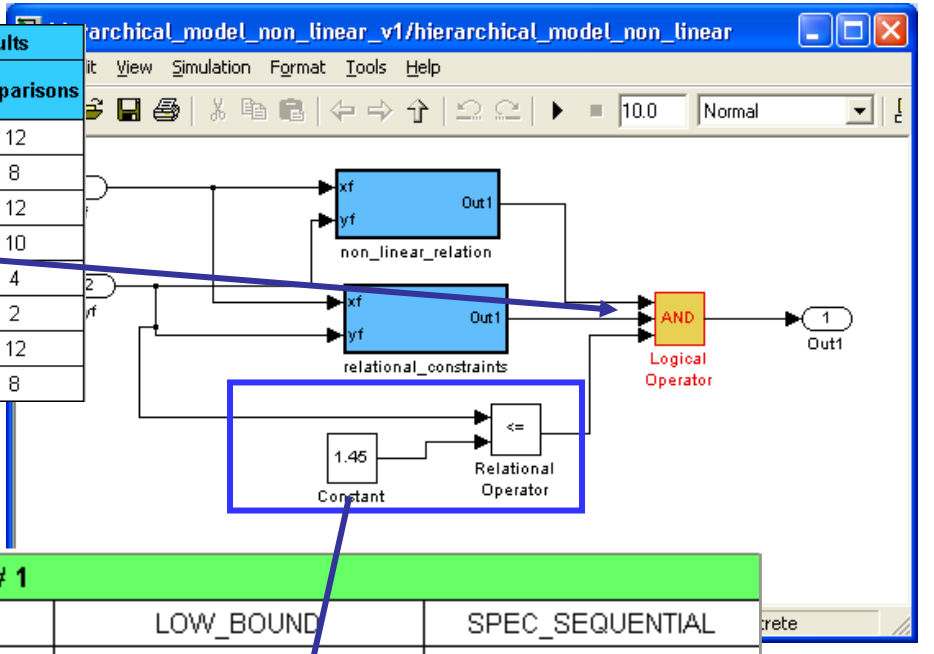
Seeded defect defines a subspace that is not within the subdomain of the other modeled subsystems



T-VEC status report links to information describing the unsatisfiable constraint and highlights the AND logical operator that is not satisfiable

Status Report

Subsystem	Compilation		Test Vectors		Coverage		Test Results	
	DCPs	Warn/Err	Vectors	Warn/Err	Untested DCPs	Warn/Err	Failures	Comparisons
child_xy	7	0/0	8	0/0	0	0/0	0	12
child_yz	7	0/0	8	0/0	0	0/0	0	8
hierarchical_model_linear	6	0/0	5	0/2	1 of 6	0/3	0	12
hierarchical_model_non_linear	5	0/0	5	0/2	1 of 5	0/3	0	10
hierarchical_model_non_linear_v1_root	1	0/0	2	0/0	0	0/0	0	4
non_linear_relation	1	0/0	2	0/0	0	0/0	0	2
parent_xy	9	0/0	10	0/0	0	0/0	0	12
relational_constraints	6	0/0	8	0/0	0	0/0	0	8



Vector Gen Error # 1			
DCP # / Fix Point Location / Fix Order	5	LOW_BOUND	SPEC_SEQUENTIAL
Reason For Error / VG Stage / Time Period	Arithmetic Operator Failure	GROUND_LEVEL_FCP	T > 0
SS File Constraint Name	non_linear_relation`RP1		
SS File Constraint Location	non_linear_relation.SS	Line # 37	Column # 24
SS File Constraint Input Domain	Occurance at UA = 10429 INTERM_Out1_oPort:booleanUint8 [1 .. 1] @ scope = 0x70002 xf_iPort:FloatType [1.1e-004 .. 2.0e+000] yf_iPort:FloatType [1.0e+000 .. 1.45e+000]		
Failed Pre-Condition Operation	cv_multiplication_of		

Mathworks' Design Verifier (DV) did not produce tests for some satisfiable test objectives where T-VEC produced test vectors

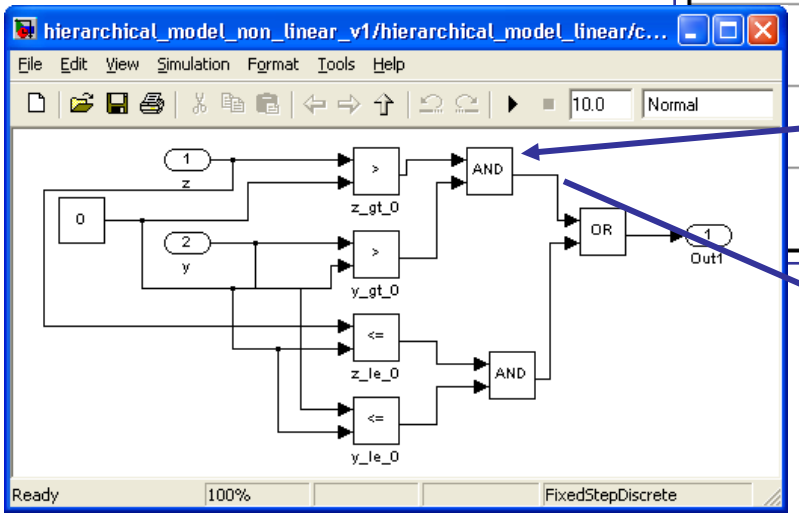
Design Verifier Report →

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
	Mcdc	Logic: MCDC expression for output with input port 1 F	Produced error	n/a
	Mcdc	Logic: MCDC expression for output with input port 2 T	Satisfied	8
	Mcdc	Logic: MCDC expression for		6



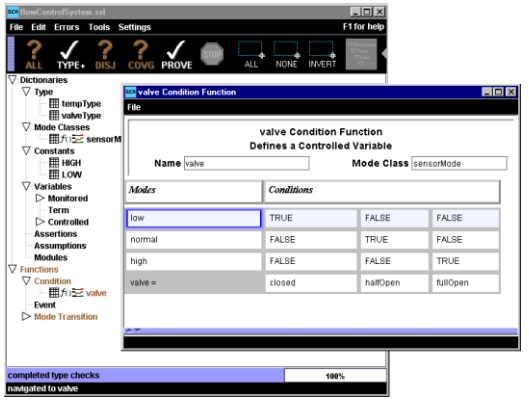
Test #	Vector #'s	_output	z_iPort	y_iPort
1	1, 3, 5, 13	1	-32768	-32768
2	2, 8, 10, 12	1	32767	32767
3	4	0	0	32767
4	6	0	32767	0
5	7	1	1	1
6	9	0	1	32768
7	11	0	-32768	1
8	14	1	0	0

Validated by senior management from The Mathworks®

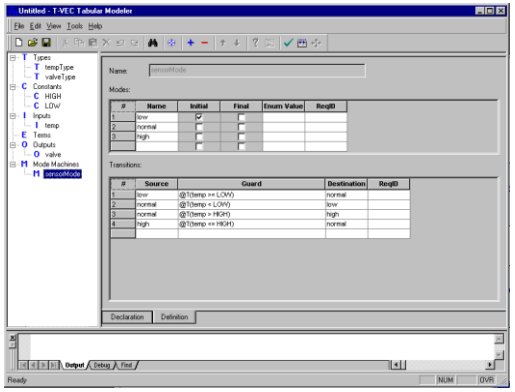
T-VEC VGS

T-VEC Tabular Modeler (TTM) extends the Software Cost Reduction (SCR) tool supporting richer data types and additional behavioral modeling constructions

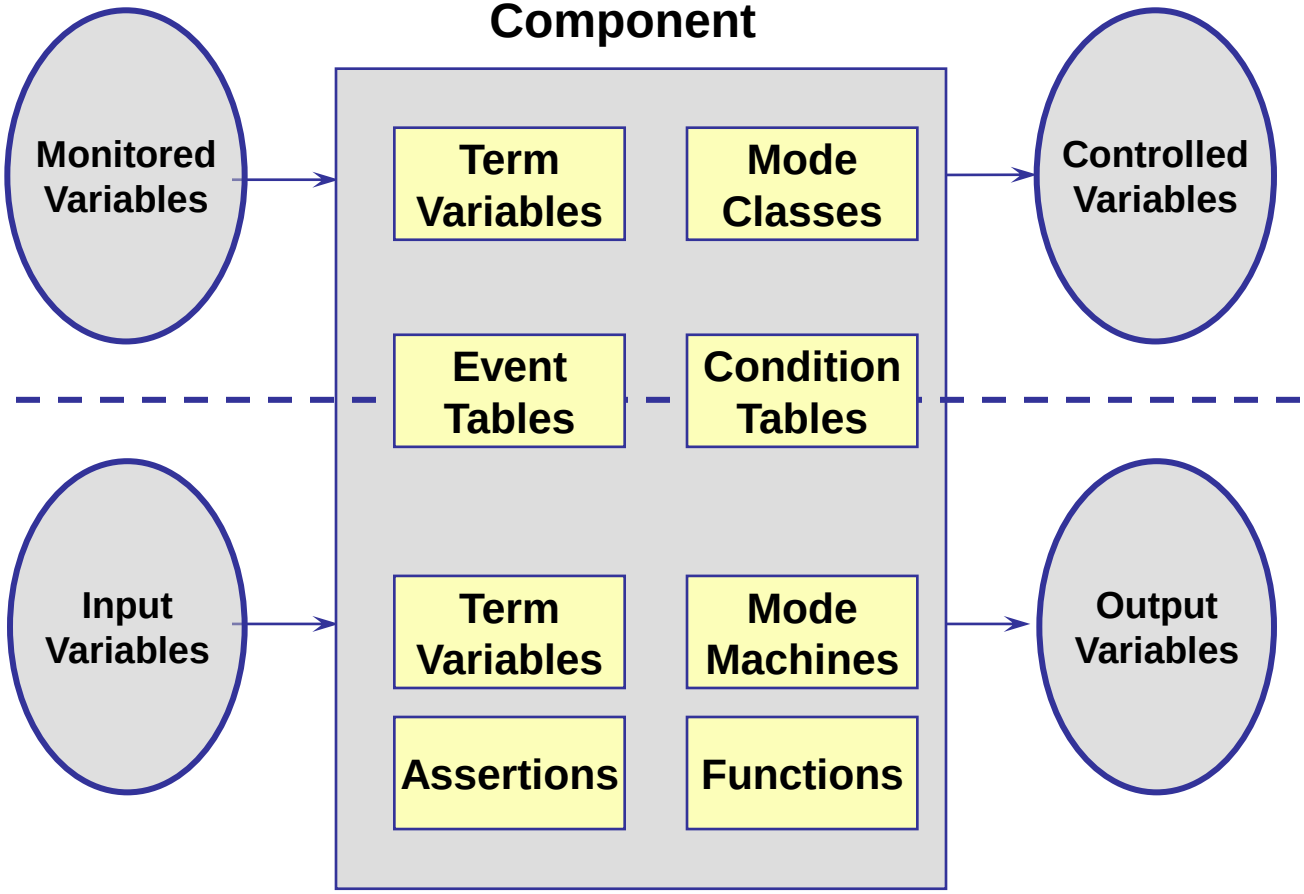
SCRtool 2.1



TTM Tabular Modeler

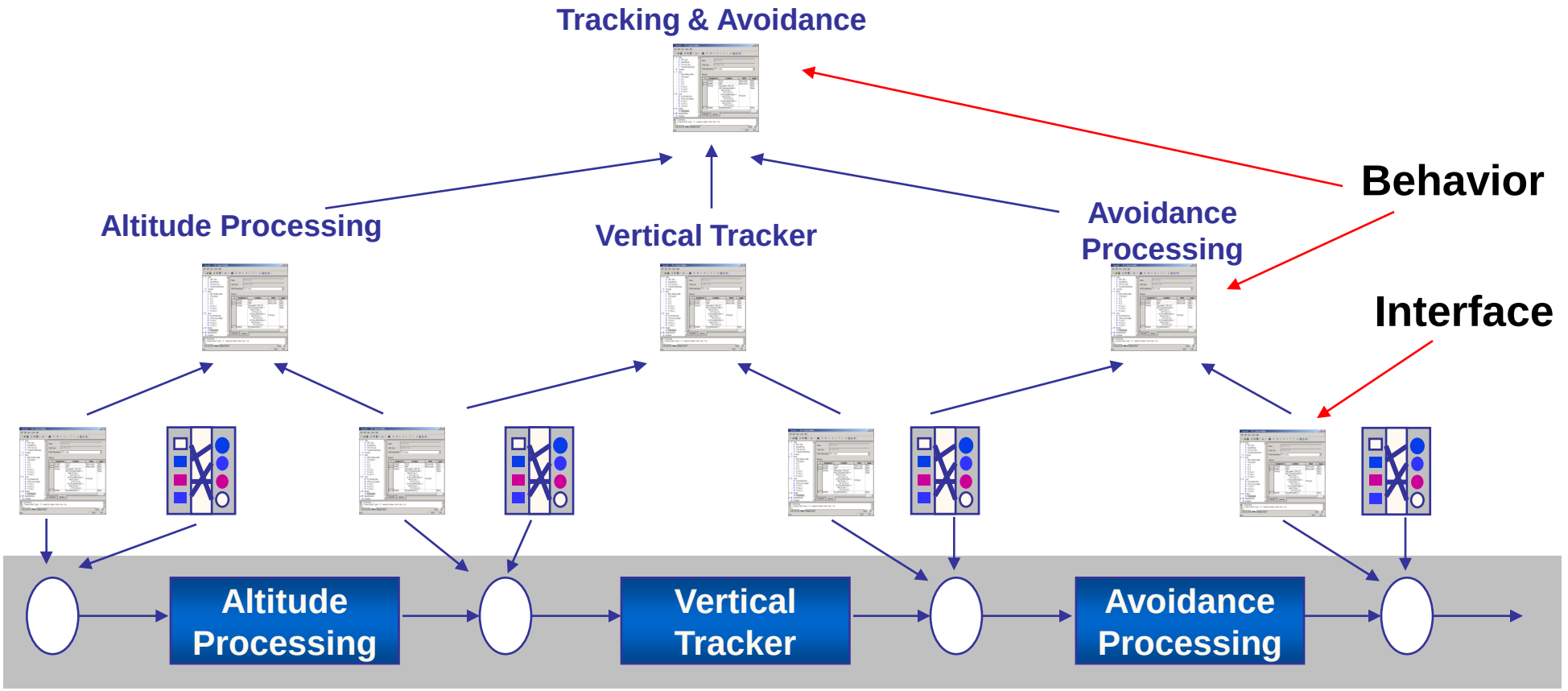


System, Software, or Component



Specified Behavior

Model references allow for inheritance, overriding, and separation of interface and behavior allowing for better model management and reuse of models



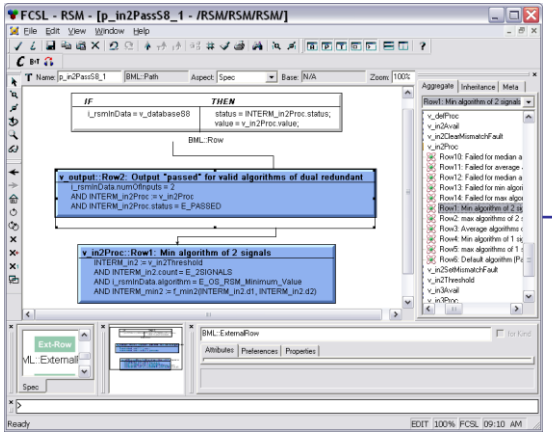
Specify the behavior for components in a separate model, which includes relevant TTM models that specify the interfaces

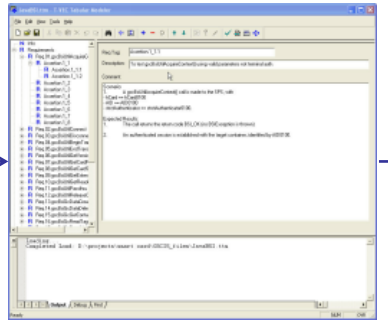
T-VEC Tabular Modeler (TTM) and Vector Generation Systems has been integrated with a Domain Specific Modeling Tool

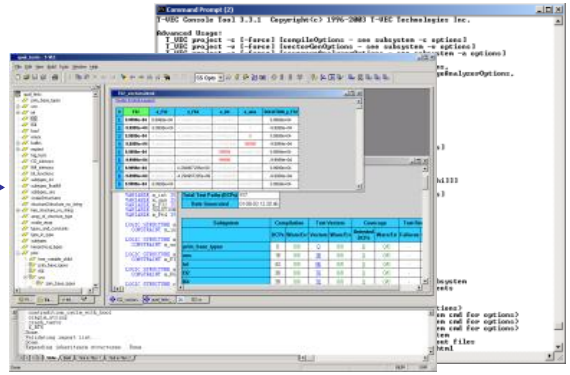
Flight Control domain-Specific Language (FCSL)

TTM

T-VEC VGS



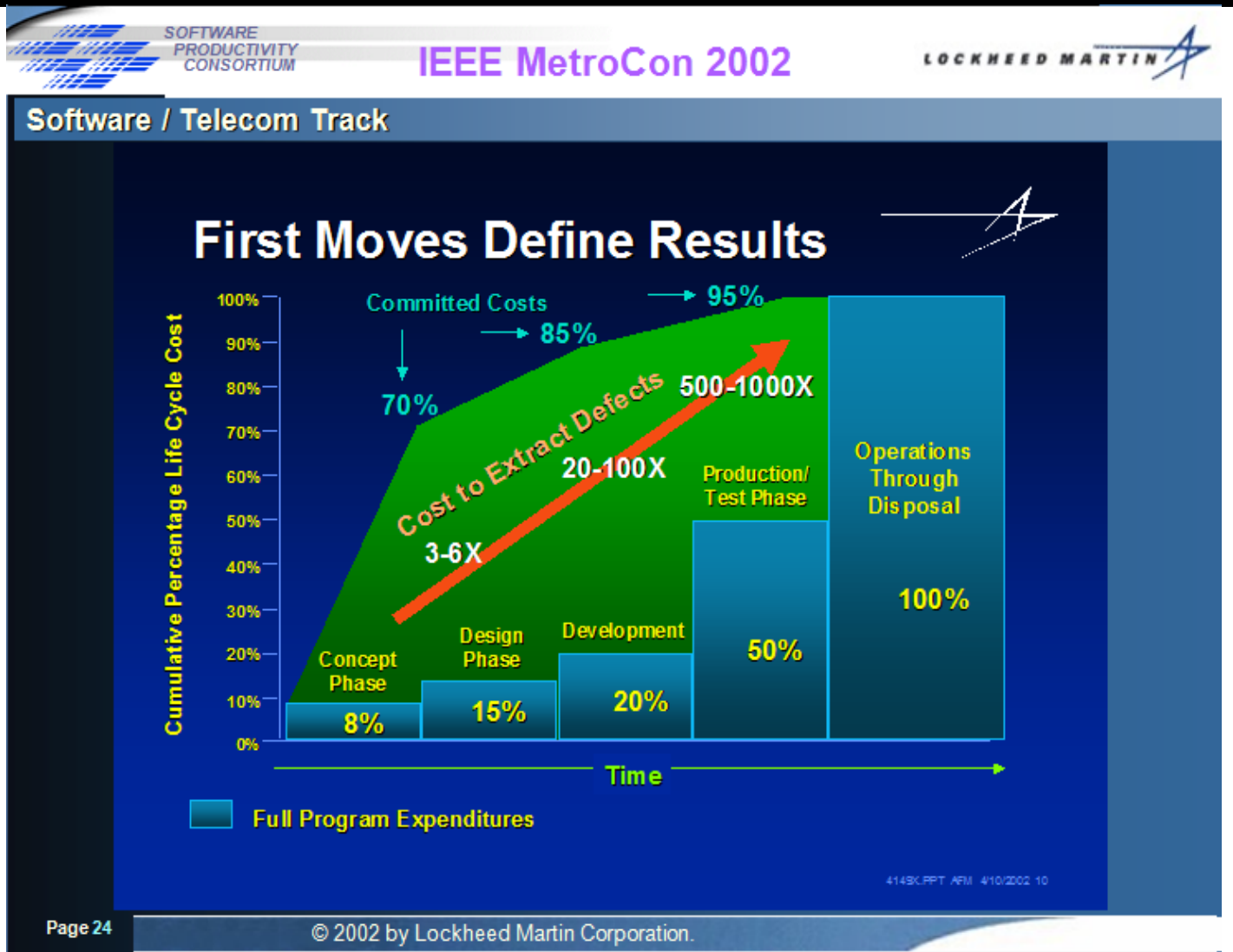




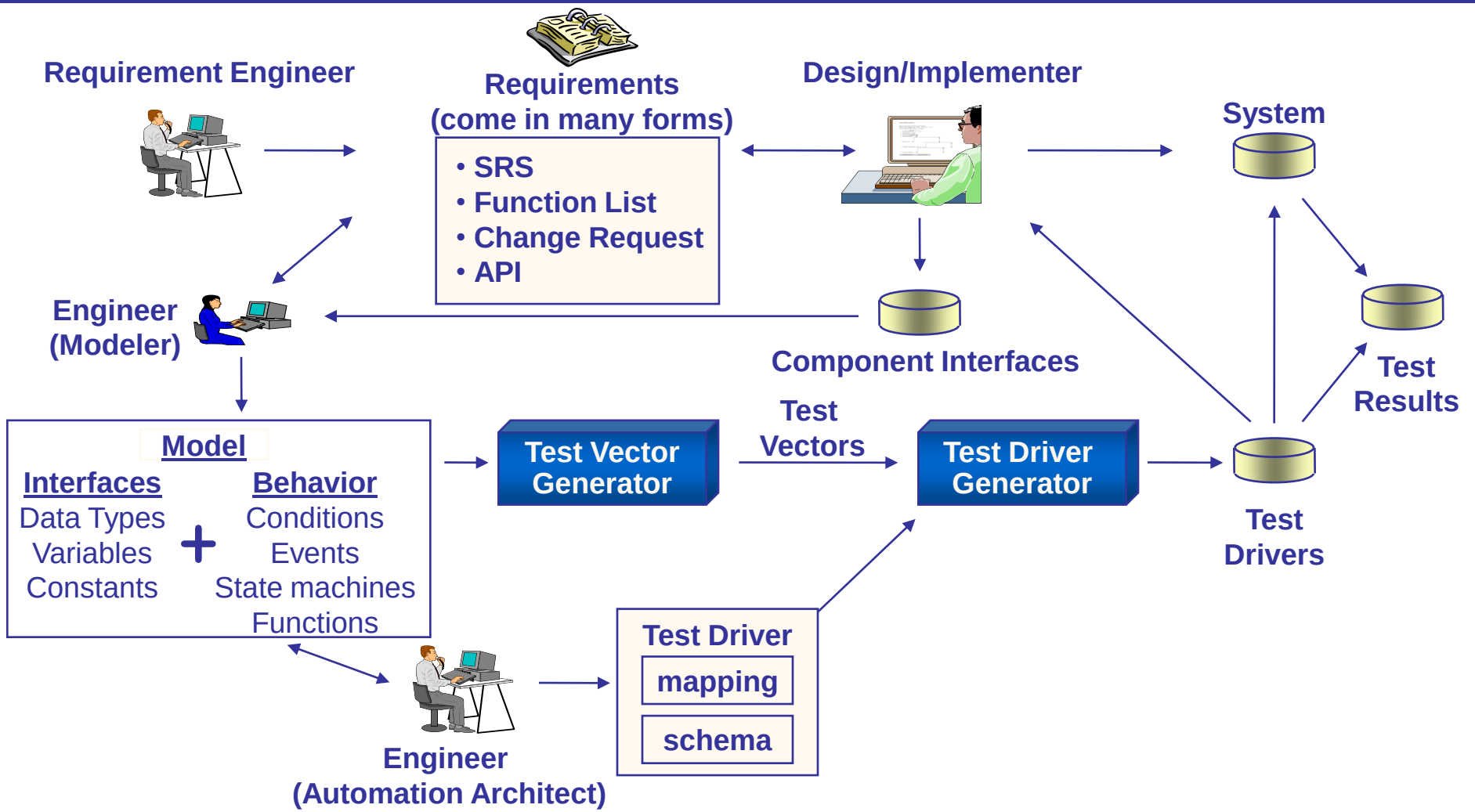
Producible Adaptive Model-based Software (PAMS) technology to the development of safety critical flight control software. PAMS has been developed under the Defense Advanced Research Projects Agency (DARPA) Disruptive Manufacturing Technologies program. Contract # N00178-07-C-2011.

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*

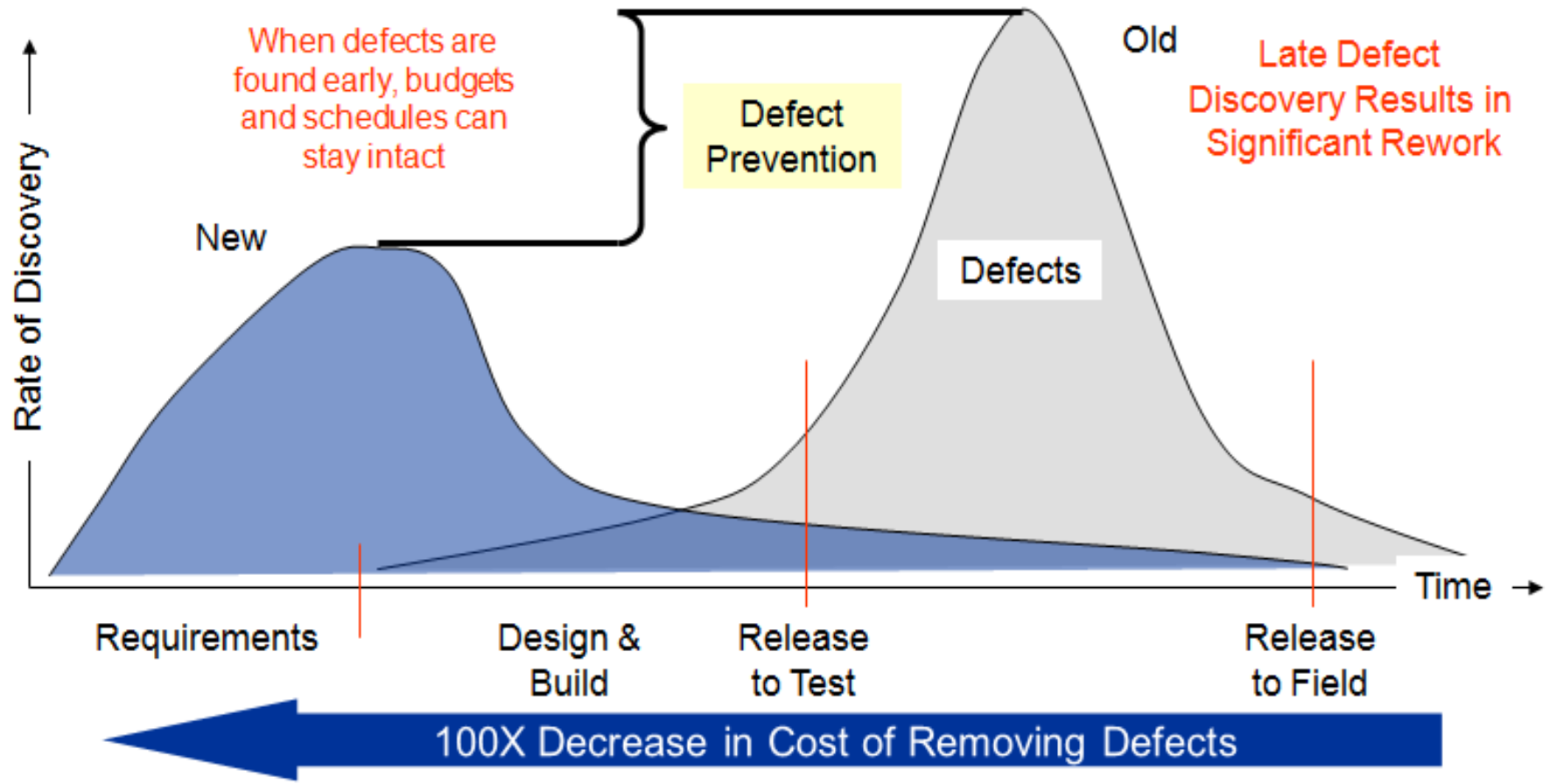
Key Point #3 – Fundamental changes in perspective have the possibility of significant cost and effort reductions



Early interface-driven approach combines requirements modeling and helps identify and correct requirements defects provides tests before implementation

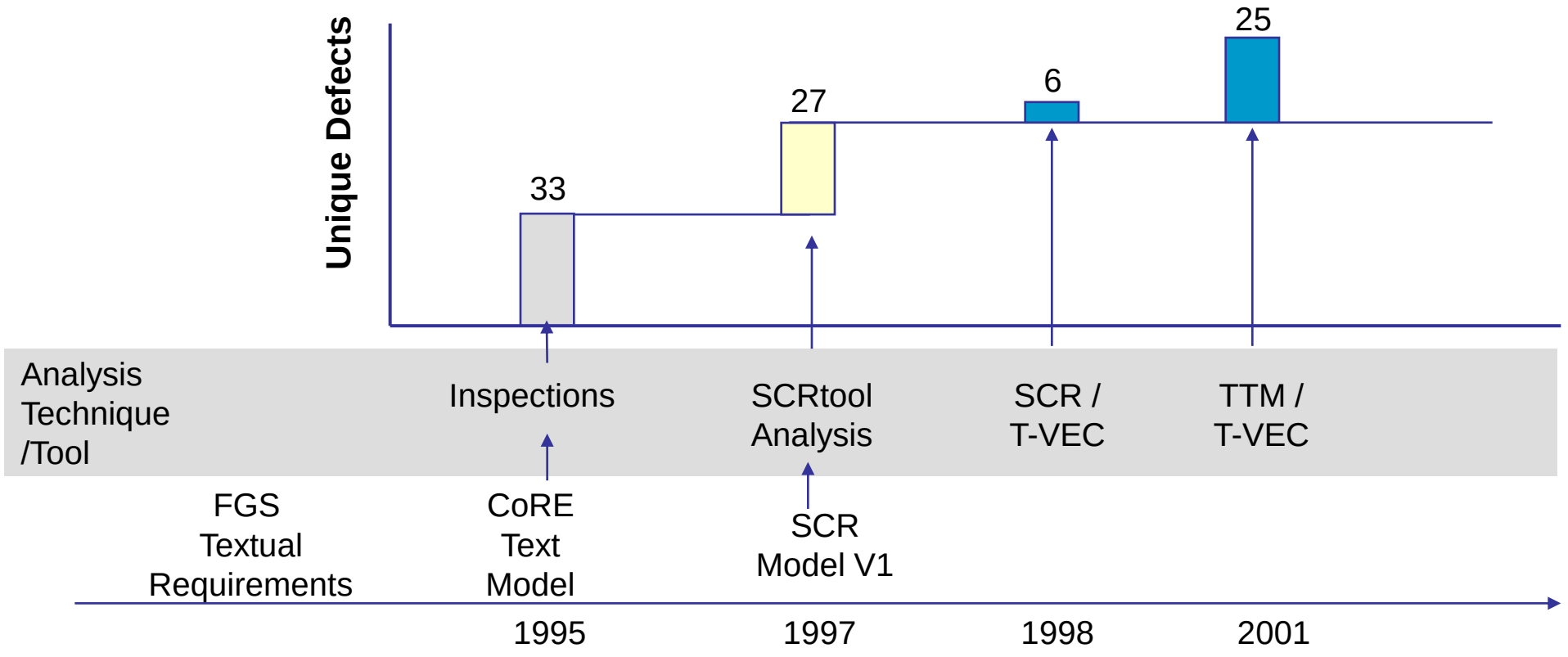


Getting the customer requirements "right" supports validation



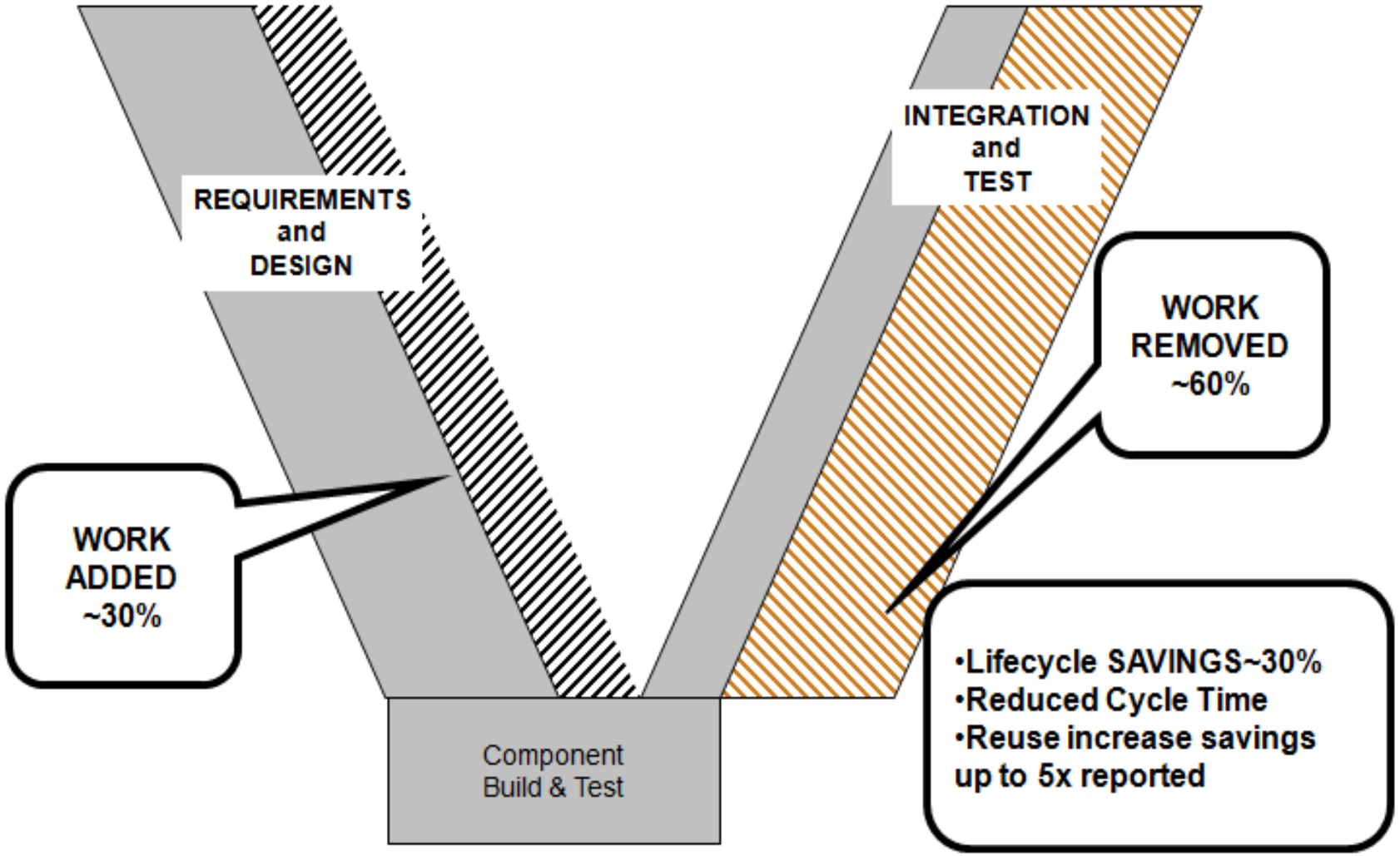
*Source: Ed Safford of Lockheed Martin, Software Technology Conference, 2000.

As the tools matured more defects in the same model where identified further illustrating that model-based automation is better than manual inspection



Rockwell Collins Pilot: Flight Guidance System (FGS) - Flight Critical Embedded System

Organization have to understand that it takes more effort up-front, but companies have evidence that it save cost and effort at the end



Closing Point – Humans alone cannot do it; automation is essential to completing the V&V for safety-critical and complex systems

- **We have to change our mindset not just the tool set**
- **It matters how the software is produced**
- **The power of the verification engine matters and for software we need to handle non-linearities**
- **We'd like to work to leverage our experience and tools to address these DARPA-hard problems**

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.