MATLAB EXPO 2017
Pratiquez la Vérification et Validation en toute sérénité

Mathieu Cuenant, MathWorks
Simulation models are primary meant to support V&V activities
Maneuverability

Instability
Integrated Design and Control of a Flying Wing Using Nonsmooth Optimization Techniques

[Yann Denieul, Joël Bordeneuve, Daniel Alazard, Gilles Taquin] - 2015
V&V techniques

Dynamic Testing

Static Analysis
Setting the scene

Pacemaker software model

Heart physical model
Algorithm design with Simulink

mode and failure management

signal processing
Checking while editing
Interactive testing
Reactive testing
Test Manager

MATLAB EXPO 2017
Requirement Proving

Step

SafetyProperty

\texttt{verify} (~ (Failure \& Mode == 1))

Step

PropertyProving

\texttt{sldv.prove} (~ (Failure \& Mode == 1))
V&V journey

Interactive testing
Reactive Testing
Coverage Analysis
Code Testing

Dynamic Testing

Edit-time checks
(Dead Logic Detection)
(Test Case Generation)

Requirement Proving
Code proving

Static Analysis

MATLAB EXPO 2017
Code Testing
Code Proving

Prove absence of run-time errors

- Polyspace Code Prover

Detect interface mismatch and concurrency issues

- Polyspace Bug Finder
Is this enough?

University of Pennsylvania Develops Electrophysiological Heart Model for Real-Time Closed-Loop Testing of Pacemakers

By Zhihao Jiang and Rahul Mangharam, University of Pennsylvania

Learn more
V&V journey

Interactive testing
Reactive Testing
Coverage Analysis
Code Testing

Edit-time checks
(Dead Logic Detection)
(Test Case Generation)

Requirement Proving
Code proving
Simulation models are primarily meant to support V&V activities.