A look to the future with Model-Based Design

Andy Grace
Vice President of Engineering
Design Automation
MathWorks Today

3 million+ users in more than 180 countries

4500+ staff in 31 offices around the world

$1B+ in 2018 revenues with 60% from outside the US

Privately held and profitable every year

Headquarters
Natick, MA USA

North America
United States

Europe
France
Germany
Ireland
Italy
Netherlands
Spain
Sweden
Switzerland
UK

Asia-Pacific
Australia
China
India
Japan
Korea

Privately held

Headquarters
Natick, MA USA

North America
United States

Europe
France
Germany
Ireland
Italy
Netherlands
Spain
Sweden
Switzerland
UK

Asia-Pacific
Australia
China
India
Japan
Korea

MathWorks Today

3 million+ users in more than 180 countries

4500+ staff in 31 offices around the world

$1B+ in 2018 revenues with 60% from outside the US

Privately held and profitable every year
Technology Megatrends Driving Automotive

1. Vehicle Electrification
2. Autonomous Driving
3. Connected Vehicles

Software everywhere
Software is reshaping the automotive industry

In the future every company will become a software company

Marc Andreessen
Founder of Netscape,
Renowned Venture capitalist
Software is reshaping the automotive industry

- Augmenting control with machine learning (BMW)
- Trailer backup assist (Ford)
- Autonomous driving (Voyage)
Agile Values

- **Individuals & Interactions** over **Process and Tools**
- **Customer Collaboration** over **Contract Negotiation**
- **Working Software** over **Comprehensive Documentation**
- **Responding to Change** over **Following a Plan**

“While there is value in the items on the right, we value the items on the left more.”  
- The Agile Alliance, 2001
Agile: Values, Principles and Practices

Agile is a mindset defined by values, guided by principles and manifested through many different practices. Agile practitioners select practices based on their needs.

~ Agile Practice Guide (PMI® and Agile Alliance®)
Typical agile development workflow

1-4 Weeks
Models == Understanding
Simulation

1-4 Weeks

Physical Prototyping
Simulation is key to Level 4-5 autonomy

Critical situations are in the long-tail*

Simulation helps achieve this improbable task

*Source: Center for Artificial Intelligence, Saarland University
Model-Based Design

Systematic use of models throughout the development process

Modeling

- Simulation

+ Automation

- Coding
- Verification

Fast repeatable tests

Fast agile development loops
Types of models

Systems

Software

Physics

Components
Physical components

Vehicle Component

Sensor Model

Communications Channel

Motor
Simscape for physical modeling

Publication-quality diagrams

Simscape modeling language

Models just run
Types of models

Systems

Software

Components

Physics
Simulink as an Integration Platform
Simulink as an Integration Platform
Simulation Integration: Analyses

Verification and Validation
- Fuel Economy
- Performance

Design Optimization
- Energy Consumption

Sensitivity Analysis
- Drivability

Virtual Calibration
- Ride & Handling

Pure EV (will update graphics)
Vehicle Dynamics (will update graphics)
Hybrid EV (will update graphics)
Automated Driving (will update graphics)
Scaling up simulations

X 1,000,000’s

Parallel simulations

Simulation Manager

Programmatic test creation
“A typical ECU contains 2000 function components that each are developed by a different person.”
SOFTWARE COMPONENTS

Working at a high-level of abstraction
Component modeling

Reusable components that can be adapted to any software system

Startup and shutdown behavior

Variant management
Types of models

Software

Physics

Components

Systems
System architecture is the #1 topic

### Breakout Topic Requests (2018)

<table>
<thead>
<tr>
<th>Topic</th>
<th>Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling System Architecture</td>
<td>75</td>
</tr>
<tr>
<td>Sensor Fusion and Tracking</td>
<td>64</td>
</tr>
<tr>
<td>Customizing Embedded Coder</td>
<td>56</td>
</tr>
<tr>
<td>Testing Simulink Models</td>
<td>55</td>
</tr>
<tr>
<td>Efficiency of Generated Code</td>
<td>51</td>
</tr>
</tbody>
</table>

### Feature Prioritization (2017)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Architecture</td>
<td>173</td>
</tr>
<tr>
<td>Code Generation</td>
<td>167</td>
</tr>
<tr>
<td>Large-scale Modeling</td>
<td>123</td>
</tr>
<tr>
<td>Verification &amp; Validation</td>
<td>106</td>
</tr>
<tr>
<td>Improved UI</td>
<td>103</td>
</tr>
</tbody>
</table>
Systems engineering

Requirements

Systems

Components
Systems engineering

Requirements

Components

System Composer
Linking top-down and bottom-up workflows
Types of models

Systems

Software

Components

Physics
Deep solutions

Controls | Signal Processing | Wireless | Vision | Robotics
MODELING & SIMULATION

Deep solutions

Automotive Products

Powertrain

Vehicle

Automated Driving

Calibration
Automotive Reference Applications

Pure EV

Lane Keeping Assist

Hybrid Powertrain

Car Vehicle Dynamics
Deep solutions for autonomous systems

- Localization
- Perception
- Planning
- Control

SLAM (18a)
Robotics System Toolbox

Semantic Segmentation (17b)
Automated Driving
System Toolbox

Path Planning (19a)
Automated Driving Toolbox

Adaptive Cruise Control (17a)
Automated Driving
System Toolbox
Deep solutions for autonomous systems

Lane Keep Assist
Model Predictive Control

Automatic Emergency Braking
Automated Driving Toolbox
MATLAB Workflow for Deep Learning:

Access Data
- Access Models
- Train
- Deploy
- Interoperability with open source networks
- Deep Network Designer App
- Inference performance

Deep Learning Toolbox
Create, analyze, and train deep learning networks

Access Data
- MUNGING/LABELING
- FUSION
- DENOISING

Preprocess
- BUILD
- BORROW

Access Models
- FROM SCRATCH
- TRANSFER

Deploy

Interoperability with open source networks
- ONNX
- PyTorch
- mxnet
- TensorFlow

Deep Network Designer App

Inference performance
- NVIDIA

Domain-specific workflow support
- Ground truth labeling apps for:
  - Video
  - Audio
  - Application-specific datastores

Network training performance
- NVIDIA
- Azure

Deployment support
- Intel
- ARM
Artificial Intelligence for your applications

- Application examples

- Object Detection Using Deep Learning
- Traffic Sign Detection and Recognition
- Pedestrian Detection
- Detecting Cars Using Gaussian Mixture Models
- Tracking Pedestrians from a Moving Car
- Waveform Segmentation using Deep Learning
Artificial Intelligence for your applications

- Application examples
- Control design

Reinforcement Learning Toolbox
Solutions for **Vision** and **Deep Learning**

- **GPU**
  - Fastest

- **FPGA / ASIC**
  - Lowest Power

- **CPU**
  - Low Cost
Model-Based Design  C/C++

- High level of abstraction
- Advanced analysis tools
- Automatic code generation
Model-Based Design

- No wrappers
- No data typing
- No data copies

C/C++ Libraries
Model-Based Design

- No wrappers
- No data typing
- No data copies

C/C++ Libraries
Modeling

Simulation

Automation

Coding    Verification
Automated Test and Verification

Find bugs: 
- Simulink Design Verifier
- Polyspace Bug Finder

Manage tests: 
- Simulink Test

Check & Coverage: 
- Simulink Check

Inspect code: 
- Simulink Coverage
- Simulink Code Inspector
Online Access for Test and Verification

CONTINUOUS INTEGRATION

AUTHENTICATION

DATA STORAGE

BUG TRACKING

Web browser
Model-Based Design

Systematic use of models throughout the development process

Modeling

Simulation

Automation

Coding Verification

Fast repeatable tests

Fast agile development loops
Who will be successful in the future?

Mechanical-centric

Model-centric

Software-centric

Comprehensive models
Simulation based testing
Generate code and automate verification
Enjoy the conference