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

Europe
- France
- Germany
- Ireland
- Italy
- Netherlands
- Spain
- Sweden
- Switzerland
- UK

Asia-Pacific
- Australia
- China
- India
- Japan
- Korea

North America
United States
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 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

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
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: Infrastructure

Data Management
Solver Technology
Vehicle Configuration
Multi-actor Scenarios
Visualization

Simulink
Simulation Integration: Analyses

- Verification and Validation
- Design Optimization
- Sensitivity Analysis
- Virtual Calibration

- Fuel Economy
- Performance
- Energy Consumption
- Drivability
- Ride & Handling

Simulink
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.”
Working at a high-level of abstraction

SOFTWARE COMPONENTS

MATLAB

Simulink

Stateflow
Component modeling

Reusable components that can be adapted to any software system

Startup and shutdown behavior

Variant management
Types of models

- Systems
- Software
- Physics
- Components

Diagram showing different types of models with examples of systems, software, physics, and components.
System architecture is the #1 topic

Breakout Topic Requests (2018)

- Modeling System Architecture: 75
- Sensor Fusion and Tracking: 64
- Customizing Embedded Coder: 56
- Testing Simulink Models: 55
- Efficiency of Generated Code: 51

Feature Prioritization (2017)

- System Architecture: 173
- Code Generation: 167
- Large-scale Modeling: 123
- Verification & Validation: 106
- Improved UI: 103
Systems engineering

Requirements

Systems

Components
Systems engineering

R 2019a
System Composer

Requirements

Components
Linking top-down and bottom-up workflows
Types of models
Deep solutions

Controls

Signal Processing

Wireless

Vision

Robotics
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 → Planning
Perception —>
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
- MUNGING/LABELING
- FUSION
- DENOISING

Preprocess
- BUILD
- BORROW

Access Models
- FROM SCRATCH
- TRANSFER

Deploy

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

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 GPU CLOUD
- Azure
- Amazon Web Services

Deployment support
- Intel
- ARM

© 2019 The MathWorks, Inc.
Artificial Intelligence for your applications

- Application examples
Artificial Intelligence for your applications

- Application examples
- Control design

Reinforcement Learning Toolbox
void AV_VecUnit_3D(float* vec) {
    float mag = sqrt(vec[0]*vec[0] + vec[1]*vec[1] + vec[2]*vec[2]);
    float scale = (1.0f/mag);
    vec[0] *= scale;
    vec[1] *= scale;
    vec[2] *= scale;
}

void AutomatedParkingPalletAlgorithm() {
    // Algorithm code goes here
}

int32_t div_s32_floor(int32_t numerator, int32_t denominator) {
    int32_t quotient;
    uint32_t absNumerator;
    uint32_t absDenominator;
    uint32_t tempAbsQuotient;
    boolean_t quotientNeedsRounding;

    if (denominator == 0) {
        return 0;
    }

    // Divide by zero handler
    if (absNumerator < 0) {
        absNumerator = static_cast<int32_t>(absNumerator) + 1;
        absDenominator = static_cast<int32_t>(absDenominator) + 1;
        quotientNeedsRounding = (numerator < 0) || (denominator < 0);
    }

    if (quotientNeedsRounding) {
        absNumerator = absNumerator;
        if (absNumerator > 0) {
            tempAbsQuotient = absNumerator / absDenominator;
            // Round down
            tempAbsQuotient = tempAbsQuotient + 1;
        }
    }

    quotient = static_cast<int32_t>(tempAbsQuotient);
    return quotient;
}
Solutions for **Vision** and **Deep Learning**

- **GPU**
  - Fastest
- **FPGA / ASIC**
  - Lowest Power
- **CPU**
  - Low Cost
Model-Based Design vs 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

Hand Code  Internal Libraries  Vendor Libraries
Model-Based Design

- No wrappers
- No data typing
- No data copies

C/C++ Libraries

Code Generation

Middleware

Hand Code

Internal Libraries

Vendor Libraries
Automated Test and Verification

Find bugs
- Simulink Design Verifier
- Polyspace Bug Finder

Manage tests
- Simulink Test

Check & Coverage
- Simulink Check
- Simulink Coverage

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

CONTINUOUS INTEGRATION

AUTHENTICATION

DATA STORAGE

BUG TRACKING

Polyspace Access

Polyspace Server R2019a

Web browser
Model-Based Design

Systematic use of models throughout the development process

Modeling

Simulation

Fast repeatable tests

Automation

Coding Verification

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