Modeling Challenges and Opportunities in Automotive Software Development

Ramesh S

ECS Process, Methods, & Tools
General Motors, GLOBAL R & D
Warren, MI
ramesh.s@gm.com

Thanks to Joseph D'Ambrosio, Paolo Giusto and several other colleagues

Disclaimer

- Based upon my perception
  - Not very long (10 years)
  - Not very close (R&D)
- For any omission, gaps and misunderstanding
  - Blame it on me

Computer vs Auto Industry Internet Joke

- Originated from a comparative statement in 1998:

  . . . If Auto industry had developed technology like PC industry

  then we could buy cars for a prize of 25$

- Evolved into a series of jokes some of which are very interesting when one looks back:
  - . . . We would have cars that would crash twice a day
  - . . . The airbag would say `are you sure’ before deploying
  - . . . You would press the start button to shut off the engine
  - . . . Occasionally your car would refuse to let you in until you simultaneously lifted the door handle, turned the key, and grabbed the radio antenna.
  - . . .

It is no longer a joke . . .

- Computer Industry appears to be taking over the auto industries
- Starting modestly in late ‘80s in ECS, Electronics & SW subsystems are slowly and steadily proliferating the vehicles
- Aggressive growth in the recent times:
  - ADAS (Advanced Driver Assist Systems) to AS (Active Safety) to AD (Autonomous Driving)
- Fully autonomous vehicles already on the roads?

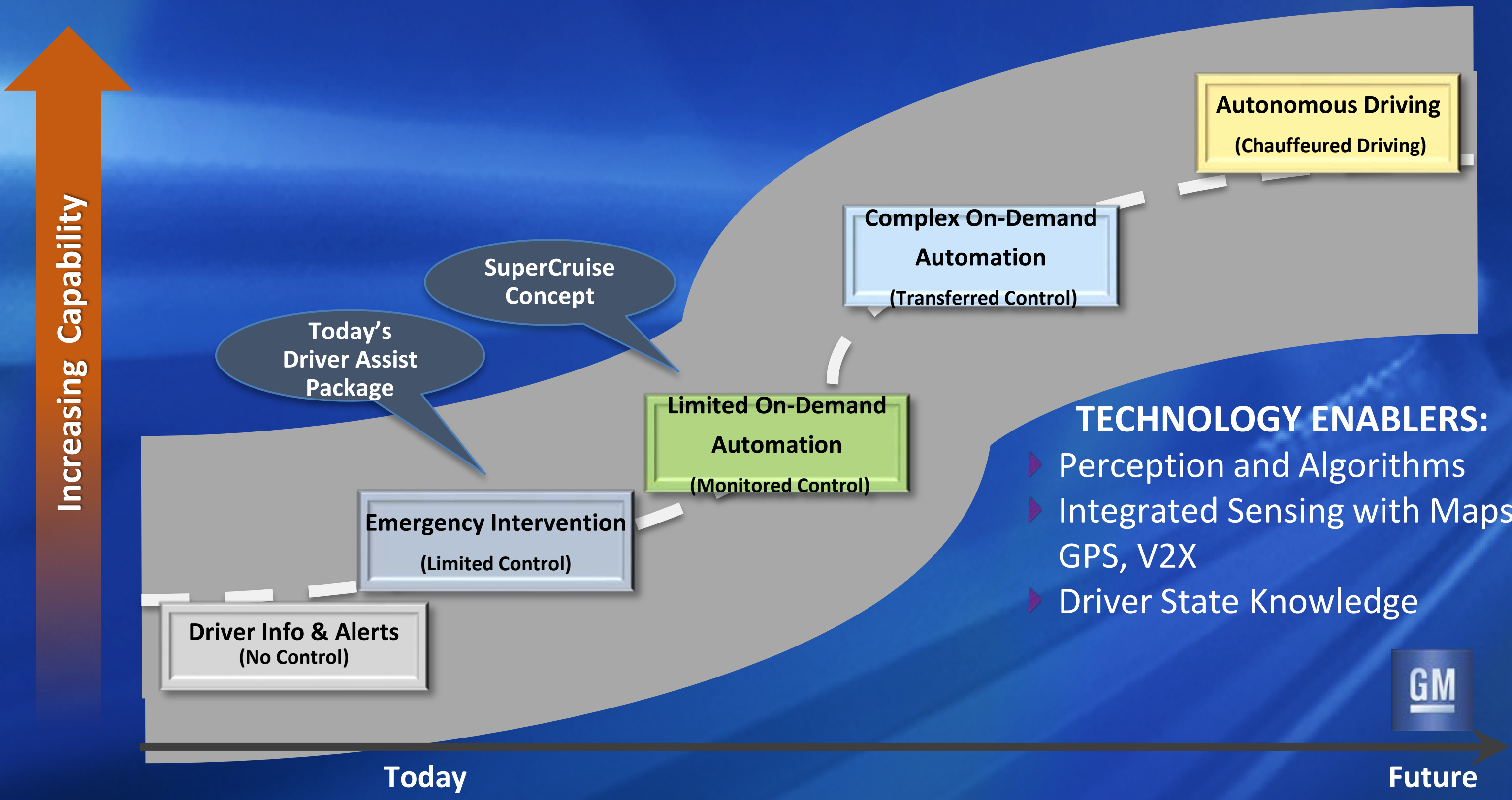
Role of MODELS

- Will this community play a (big) role in this?
- Which version of UML would be most helpful?
- How much of modeling (and analysis) required for bringing these vehicle out onto the roads?
- Is there a considerable body of work, methods and tools from this community which can be put to use?
- Which directions to go?

Outline

- Complexity of Automotive Systems & Development
- Current Modeling practices in the Industry
    - MBD of components mature
    - System level modeling being introduced
    - More of modeling and less of analysis
- Models capture possibilities and design choices – Design Space Exploration
- Required Enhancements
- Challenges in realizing them

GM ROADMAP TO AUTOMATED DRIVING



Increasing Capability

Driver Info & Alerts
(No Control)

Emergency Intervention
(Limited Control)

Today's
Driver Assist
Package

SuperCruise
Concept

Limited On-Demand
Automation
(Monitored Control)

Complex On-Demand
Automation
(Transferred Control)

Autonomous Driving
(Chauffeured Driving)

Today

Future

TECHNOLOGY ENABLERS:

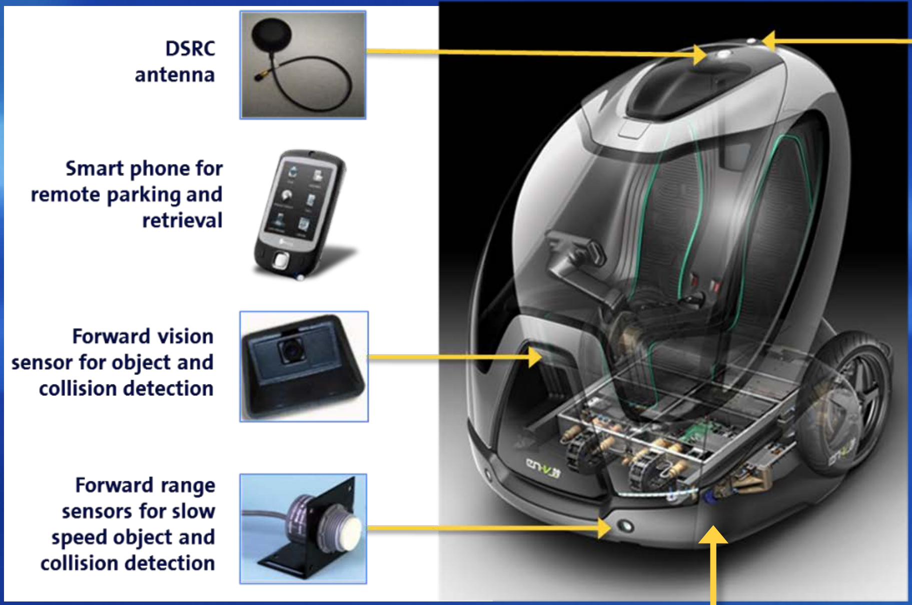
- Perception and Algorithms
- Integrated Sensing with Maps, GPS, V2X
- Driver State Knowledge

EARLY GM AUTOMATED VEHICLES

NOVEMBER 3, 2007: “BOSS” WINS DARPA URBAN CHALLENGE

AUTONOMOUS DEMOS

Personal Urban Mobility: EN-V
(Electric, Networked Vehicle)

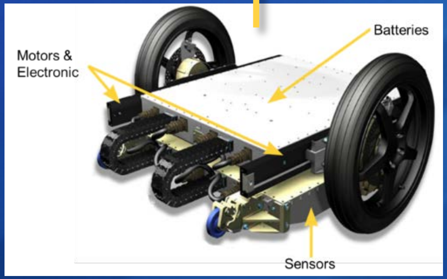

DSRC antenna

Smart phone for remote parking and retrieval

Forward vision sensor for object and collision detection

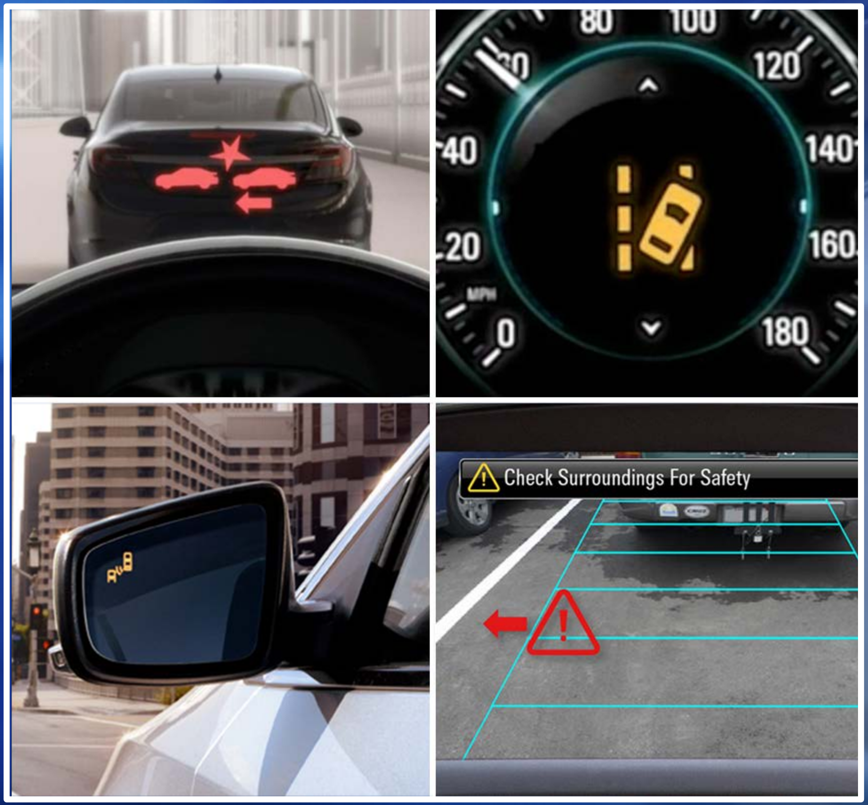
Forward range sensors for slow speed object and collision detection

GPS antenna



AUTONOMOUS with
BY-WIRE PROPULSION AND
CHASSISHARDWARE

TODAY'S TECHNOLOGIES

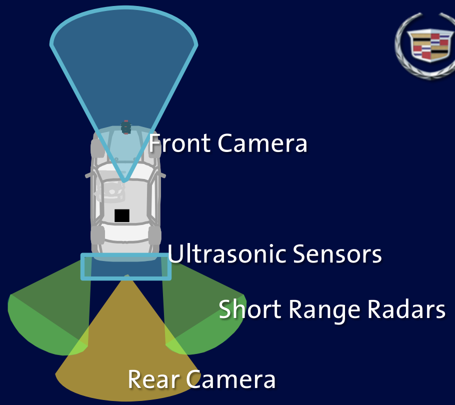


Check Surroundings For Safety

CADILLAC DRIVER ASSISTANCE / ACTIVE SAFETY

Package 1 – "Driver Awareness Package"

Cadillac ATS
Cadillac XTS
Cadillac SRX



Safety Alert Seat

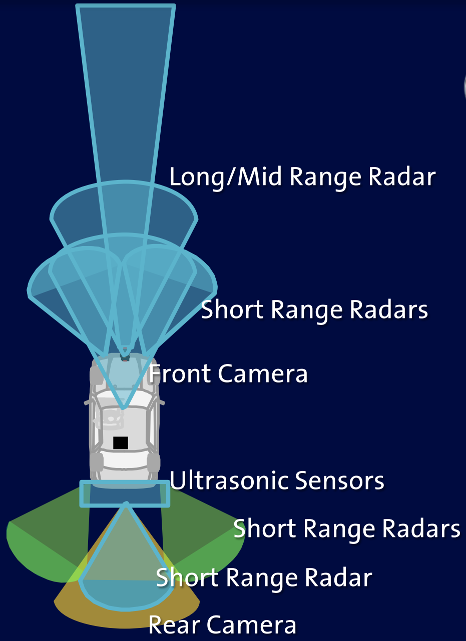
- Lane Departure Warning
- Forward Collision Alert
- Side Blind-Zone Alert
- Rear Cross-Traffic Alert
- Haptic Safety Alert Seat Feedback

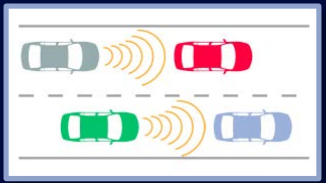
Also includes:

- Rear Vision Camera
- Front & Rear Park Assist

Package 2 – "Driver Assist Package"

Cadillac ATS
Cadillac XTS
Cadillac SRX

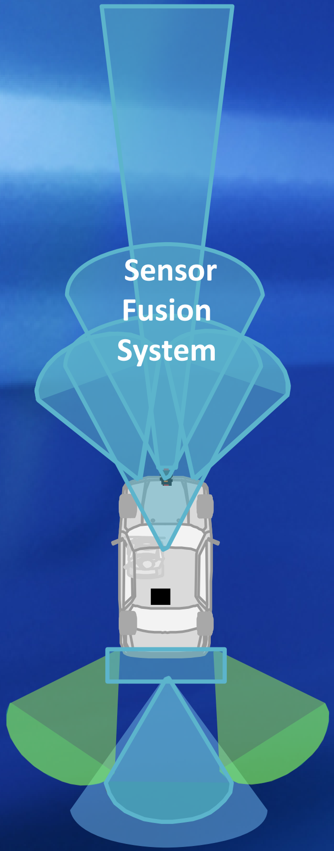




Package 1 plus:

- Full Speed-Range ACC (Stop w/Go Notifier)
- Auto Collision Preparation (includes Collision Imminent Braking)
- Low-Speed Front/Rear Automatic Braking (Emergency Braking to Avoid Contact)

CADILLAC DRIVER ASSIST

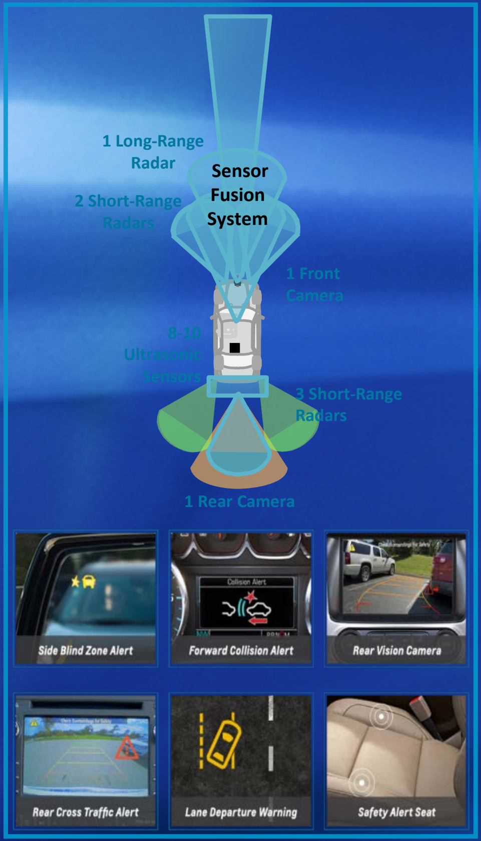


Sensor Fusion System

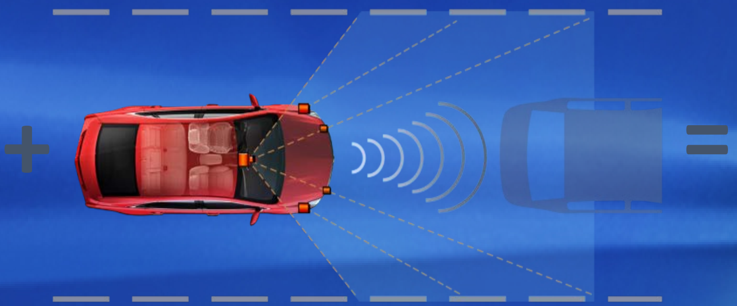
ATS
XTS
SRX
CTS

Cadillac to introduce Super Cruise on ALL-NEW ct6

ACTIVE SAFETY



AUTOMATED STEERING & LANE FOLLOWING



HOW IT WORKS

LANE FOLLOWING: Using a combination of GPS and optical cameras, Super Cruise watches the road ahead and adjusts steering to keep the car in the middle of its lane.

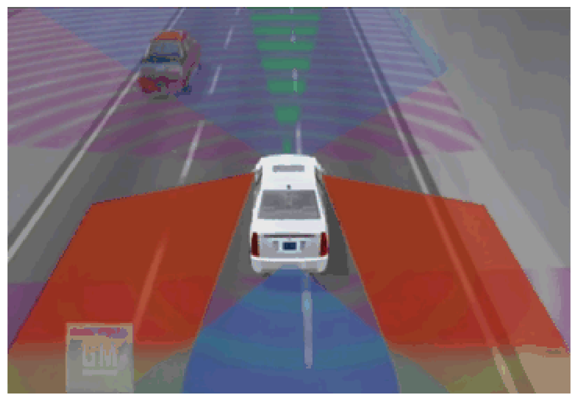
COLLISION AVOIDANCE: A long-distance radar system detects vehicles more than 300 ft. ahead. The vehicle will automatically accelerate or apply the brakes to maintain a preset following distance.

CADILLAC SUPER CRUISE

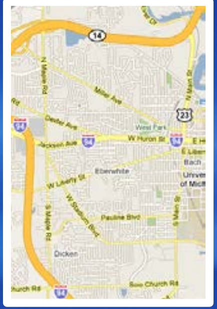
Prevents 10 K deaths, Saves 250 Billion Dollars – Boston Consulting Co.

INTEGRATED SYSTEMS APPROACH

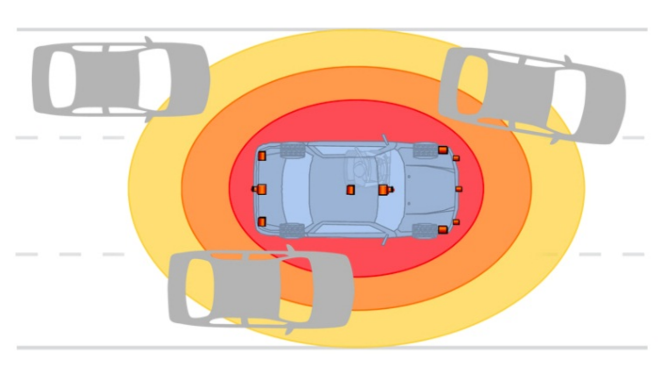
360° SENSING



MAPS/GPS



SENSOR FUSION



V2V/V2I INTEGRATION

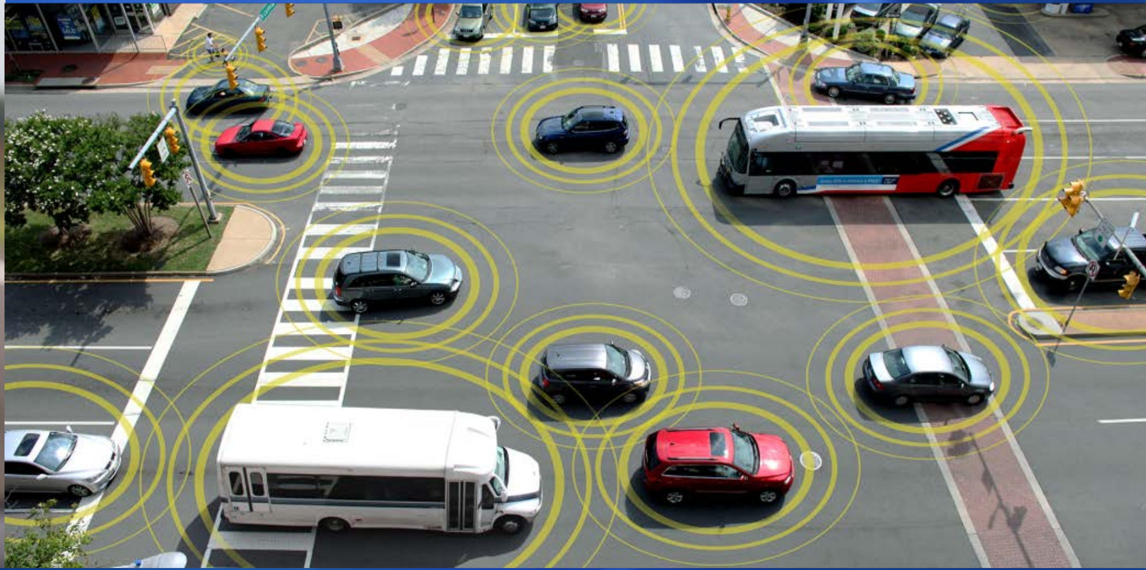
GM Speeds Up with OnStar 4G LTE

- Built-in Wi-Fi hotspot
- Connect multiple mobile devices at once
- Faster, more reliable connection
- Connect to vehicle remotely
- On more than 30 GM vehicle models

**NEW FOR 2016: PROGNOSTICS;
ANDROID AUTO/APPLE CAR PLAY**

V2X to DEBUT on 2017 Cadillac CTS

Technology allows cars to communicate with each other (V2V), the infrastructure (V2I), and pedestrians (V2P)



V2X Safety Applications

EARLY APPLICATIONS

Vehicle to Vehicle (V2V)	Vehicle to Infrastructure (V2I)	Vehicle to Pedestrian (V2P)
Hazardous Vehicle Warning	Curve Speed Warning	Pedestrian Awareness
Emergency Electronic Brake Lamps	Work Zone Warning	Cyclist Awareness
Road Condition Warning	Traffic Signal Violation Warning	
Cross Traffic Alert		

FUTURE OPPORTUNITIES

2017+

Basic Warning

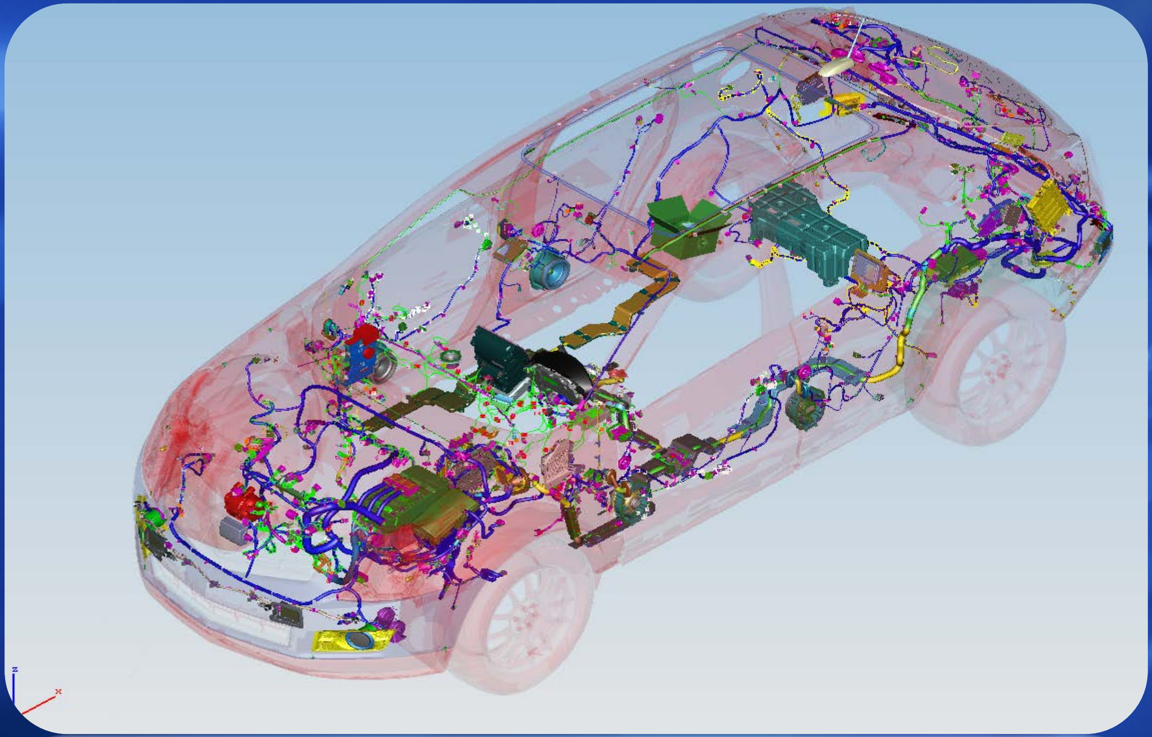
2020+

Active Safety Warning

2022+

Active Safety Control
No Other Sensors

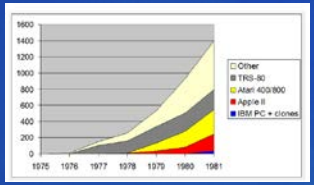
Electronics, Control & SW Design



History of GM Automotive Computing

- First Embedded Controllers
  - 1977 – First GM production automotive microcontroller
    - Electronic spark timing
  - 1981 – All GM North American vehicles use microcontroller-based engine controls
    - 3.9M vehicles total, 22K ECMs per day manufacturing rate
    - 50,000 lines of assembly code, MC6800 – 8-bit 2 Mhz,
    - Comparison against PC industry

PC Sales (in 1000s)



The holy trinity: Commodore PET, TRS-80 Model I, and the Apple][

- Today
  - 40-70 microcontrollers per vehicle
  - 400K Lines of C Code for an engine control application
  - 64Mb flash file system for infotainment application

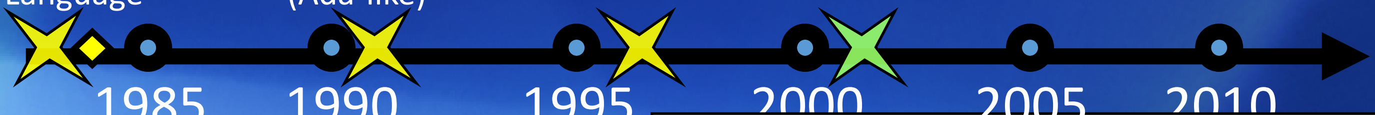
GM Embedded Software History

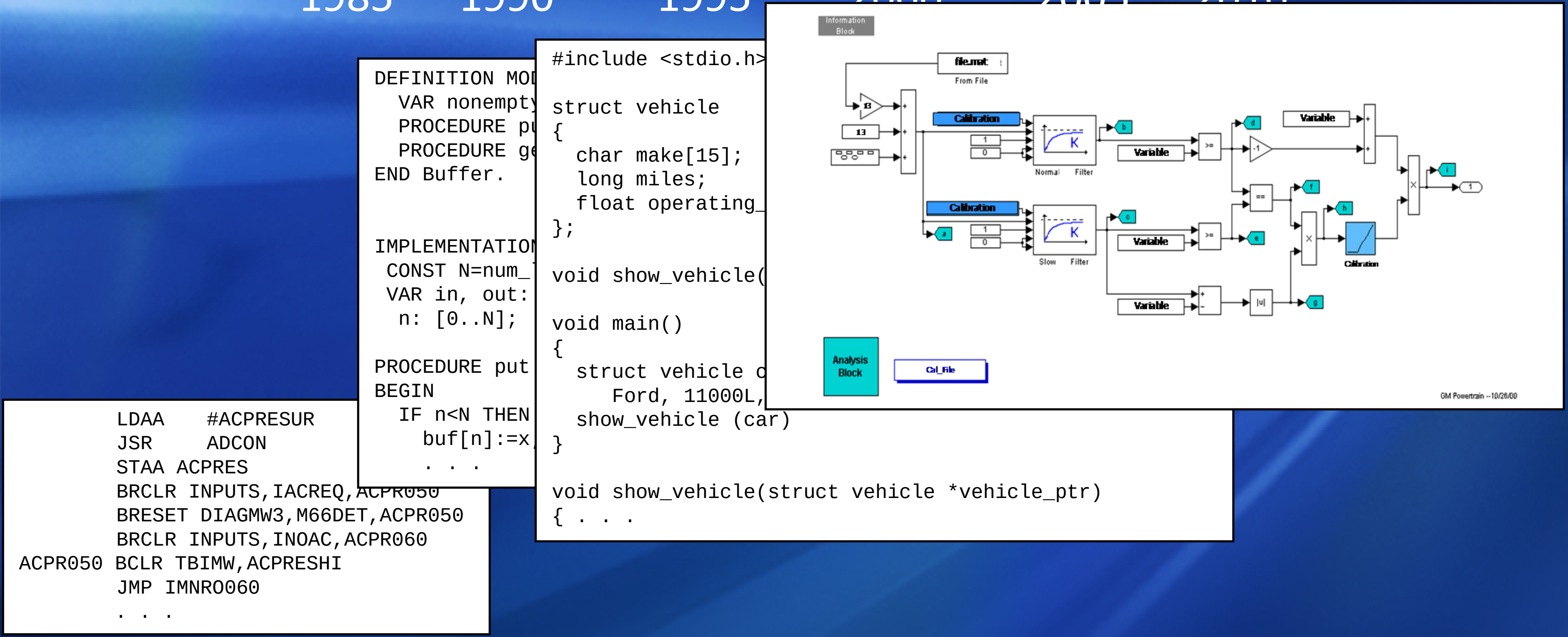
Assembly Language

Modula-GM (Ada-like)

ANSI C

Model-based Development





```
        LDAA    #ACPRESUR
        JSR     ADCON
        STAA ACPRES
        BRCLR INPUTS,IACREQ,ACPR050
        BRESET DIAGMW3,M66DET,ACPR050
        BRCLR INPUTS,INOAC,ACPR060
ACPR050 BCLR TBIMW,ACPRESHI
        JMP IMNRO060
        . . .
```

```
DEFINITION MOD
  VAR nonempty
  PROCEDURE pu
  PROCEDURE ge
END Buffer.


IMPLEMENTATION
 CONST N=num_
 VAR in, out:
  n: [0..N];

PROCEDURE put
BEGIN
  IF n<N THEN
    buf[n]:=x
     . . .
```

```
#include <stdio.h>

struct vehicle
{
  char make[15];
  long miles;
  float operating_
};

void show_vehicle(

void main()
{
  struct vehicle c
     Ford, 11000L,
  show_vehicle (car)
}

void show_vehicle(struct vehicle *vehicle_ptr)
{ . . .
```

Automotive Systems of Systems

- Deeply Embedded
    - Real Time, Possibly Safety Critical
    - Examples: Electronic Power Steering, Electronic Brake Controls, Powertrain, Active safety
    - Development Tools: Simulink/Stateflow
    - Future SW Architecture: AUTOSAR
- Moderately Embedded
    - Loosely Real Time
    - Example: Body Control, Instrument Panel, Heating/Cooling
    - Development Tools: e.g., Rhapsody
    - Future SW Architecture: AUTOSAR
- Lightly Embedded
    - Non Real Time, but may include data streaming; Security is important
    - Example: Infotainment Systems
    - Future SW Architecture: e.g., QNX/Linux, ANDROID, …

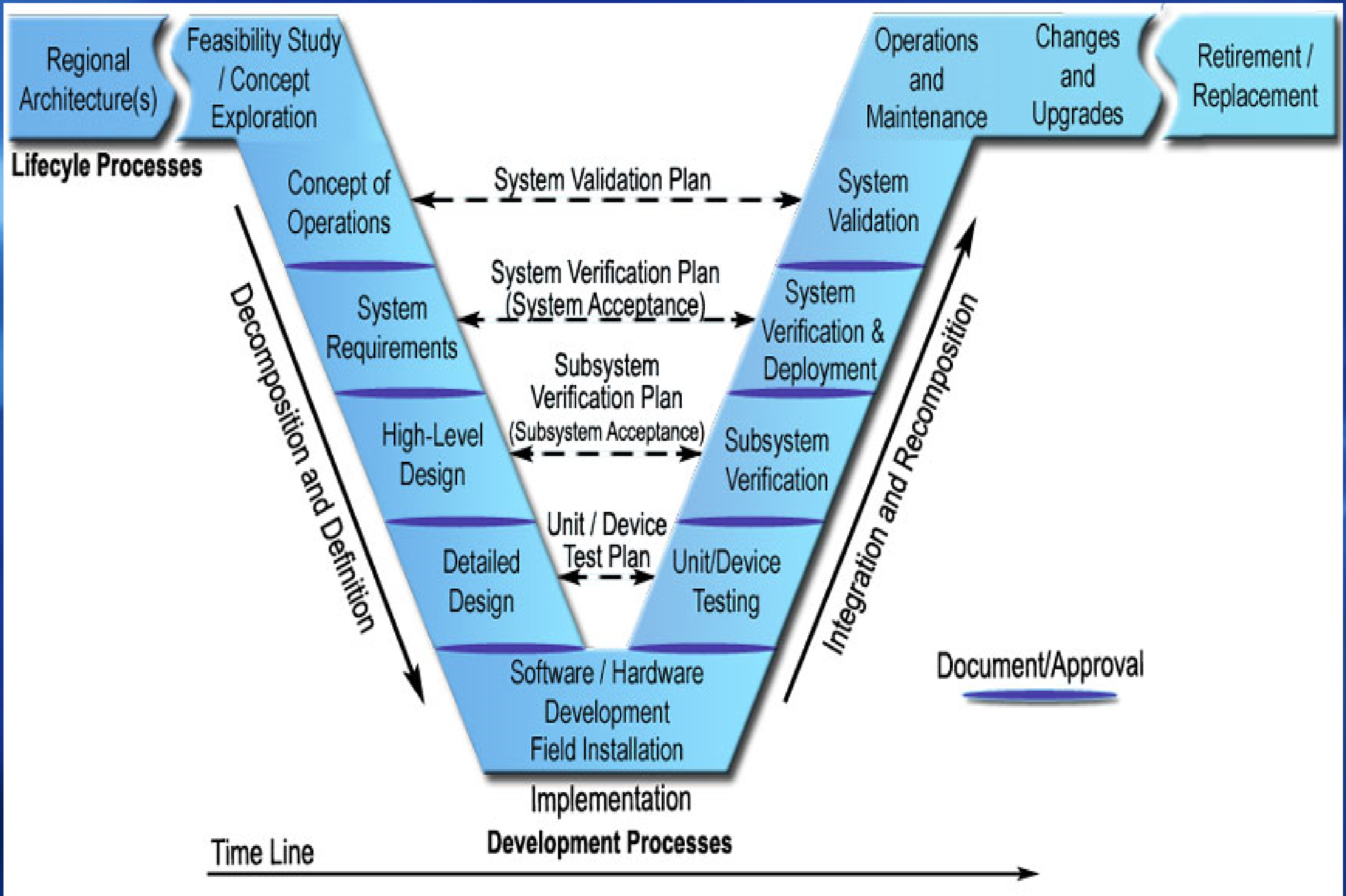
Classes of Embedded Systems

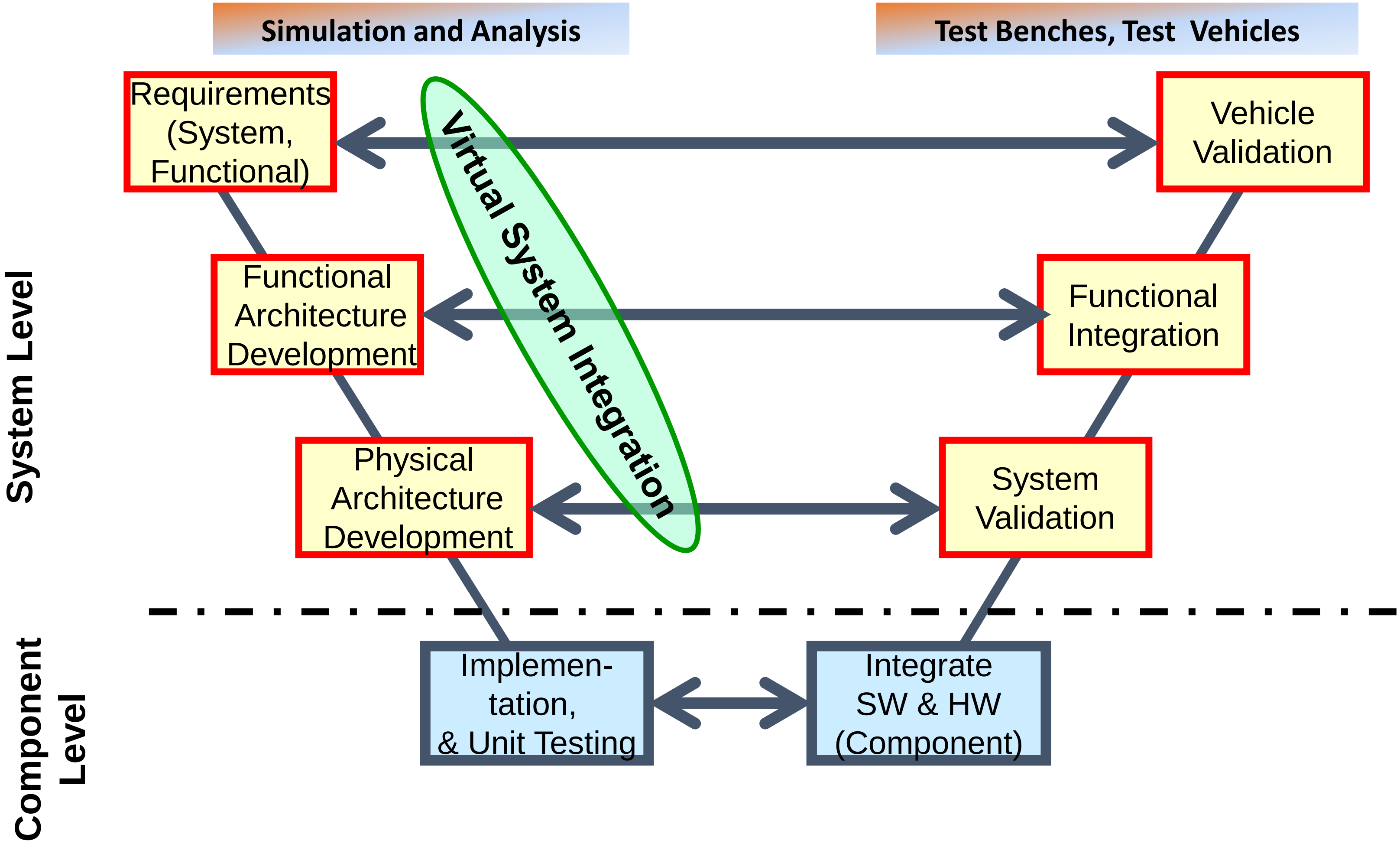
- Closed-Loop Control Systems
  - Based upon control system theory (e.g., PID control)
  - Examples: Steering systems, braking systems, propulsion systems
  - GM Tools: Simulink/Stateflow
- State-Based Systems
  - Based upon state transition diagrams
  - Example: Body control
  - GM Tools: Rhapsody

Classes of Embedded Systems

- Non Safety Critical – no potential to cause harm
  - Detect fault, save diagnostic trouble code, possibly alert driver
  - Tools: DFMEA, Requirements-Based Testing, …
- Safety Critical – potential to cause harm; timing properties are important
  - Fail Safe – detect fault, shut down within required fault response time, warn driver
  - Fail Operational – detect fault, continue to operate, possibly in a degraded mode, warn driver
  - Tools: Preliminary Hazard Analysis, Safety Concept, DFMEA, Fault Tree Analysis, Requirements Analysis, … Safety Case

V-Model of Development

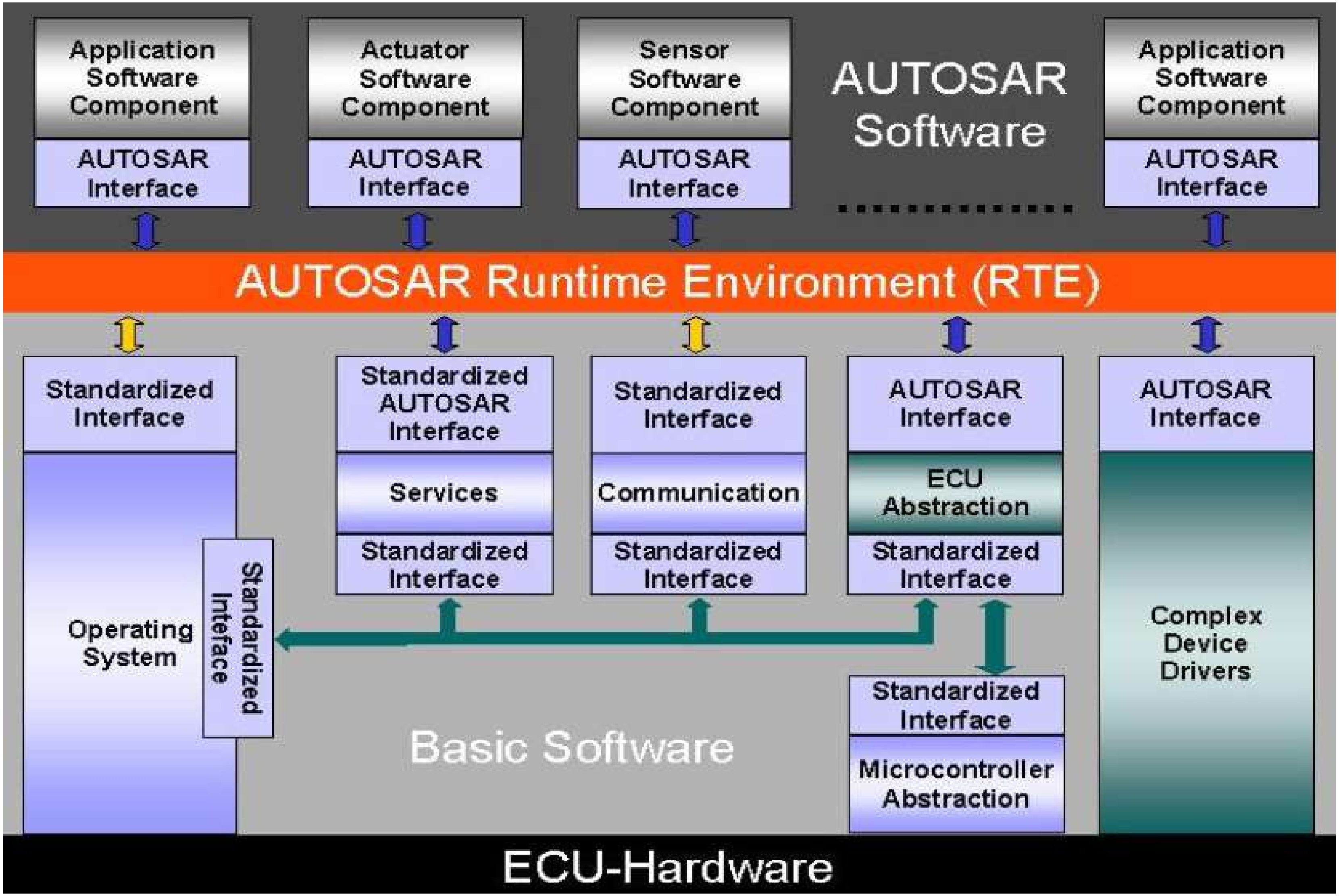


Simulation and Analysis
Test Benches, Test Vehicles
Requirements (System, Functional)
Vehicle Validation
Virtual System Integration
Functional Architecture Development
Functional Integration
Physical Architecture Development
System Validation
System Level
Implemen-tation, & Unit Testing
Integrate SW & HW (Component)
Component Level

Overall Development Strategy

- Platform based Development of Systems
  - Common Global Electrical Architecture
  - Common Software Architecture
    - Internal as well as Industry-wide Autosar standard
- Product-line Oriented Development
- Model based Development
- System Engineering Approach
  - Early and Elaborate Safety Analysis
  - Serious Requirement Engineering

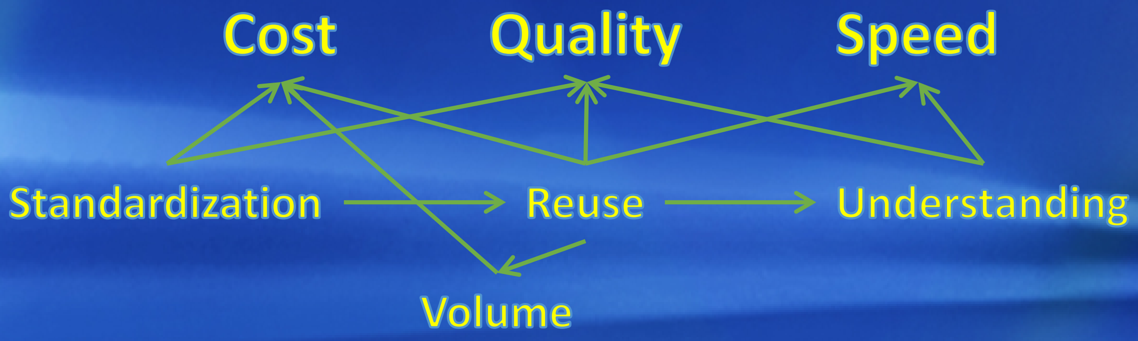
AUTOSAR Architecture



Product-line Engineering

- GM has one of the most complex systems and software product line engineering challenges in the world
  - 3000 contributing engineers
  - 300 hierarchical subsystems
  - Thousands of variant features
  - 100 Million lines of code
  - Millions of product instances per year
  - Tens-of-thousands of unique product variants
  - Dramatic increase in variation due to new propulsion systems and active safety
  - Global diversity in legislative regulations
  - Extreme economic and competitive pressures
  - Product line and feature set evolves annually
  - 15 concurrent development streams

System Design Motivation



Cost

Quality

Speed

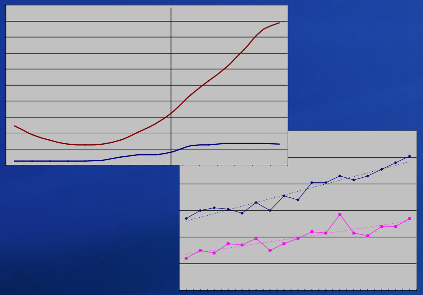
Standardization

Reuse

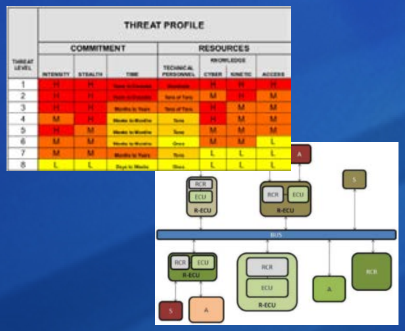
Understanding

Volume

Growing Complexity



Safety and Security



THREAT LEVEL	THREAT PROFILE						
	COMMITMENT			RESOURCES			
					KNOWLEDGE		
	INTENSITY	STEALTH	TIME	TECHNICAL PERSONNEL	CYBER	KINETIC	ACCESS
1	H	H	Years to Decades	Hundreds	H	H	H
2	H	H	Years to Decades	Tens of Tens	M	H	M
3	H	H	Months to Years	Tens of Tens	H	M	M
4	M	H	Weeks to Months	Tens	H	M	M
5	H	M	Weeks to Months	Tens	M	M	M
6	M	M	Weeks to Months	Ones	M	M	L
7	M	M	Months to Years	Tens	L	L	L
8	L	L	Days to Weeks	Ones	L	L	L

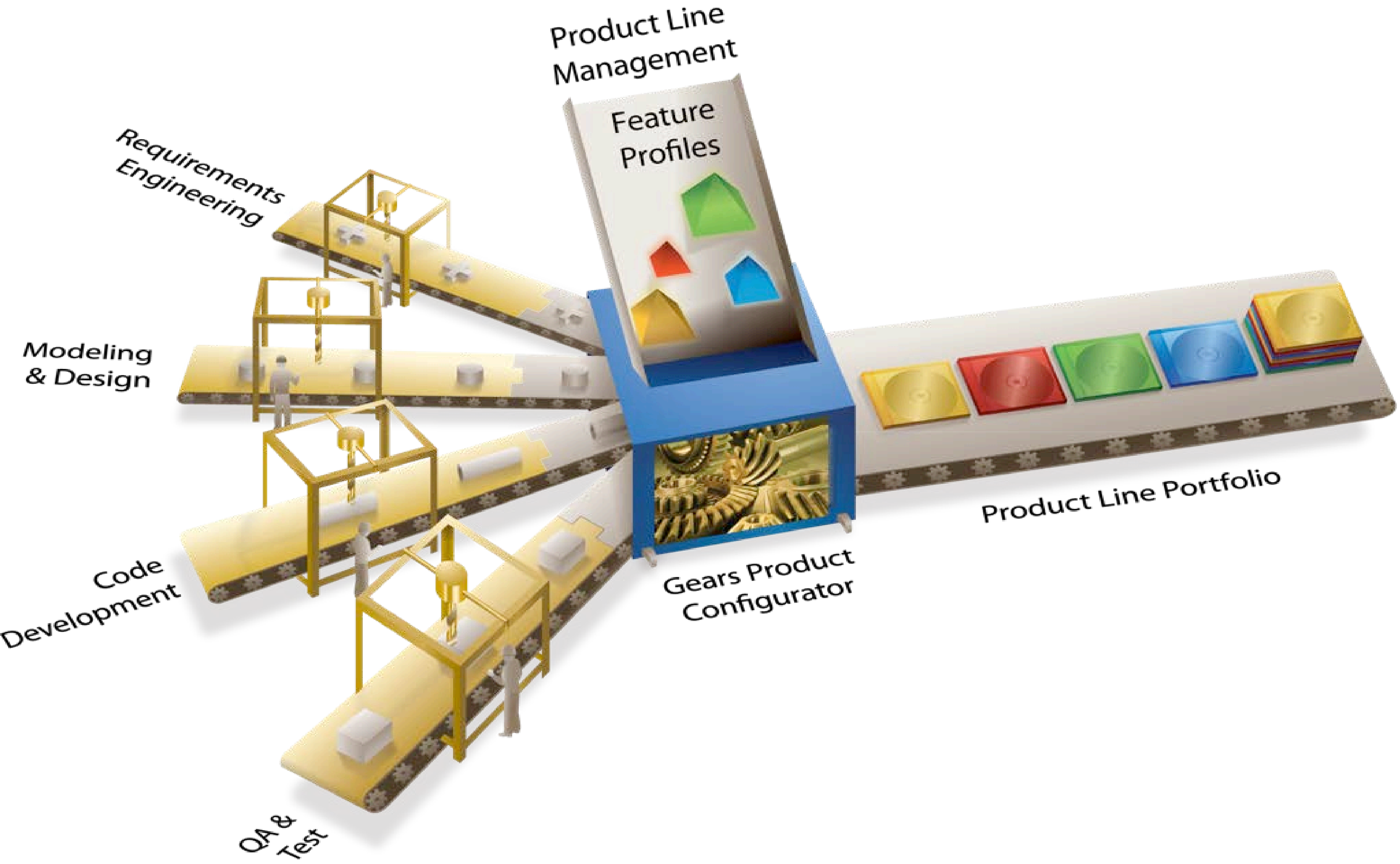
Global Footprint

GM Enables massive Reuse through Software Product Lines

- A Product Line is a set of systems sharing a common, managed set of features that are developed from a common set of core assets in a prescribed way

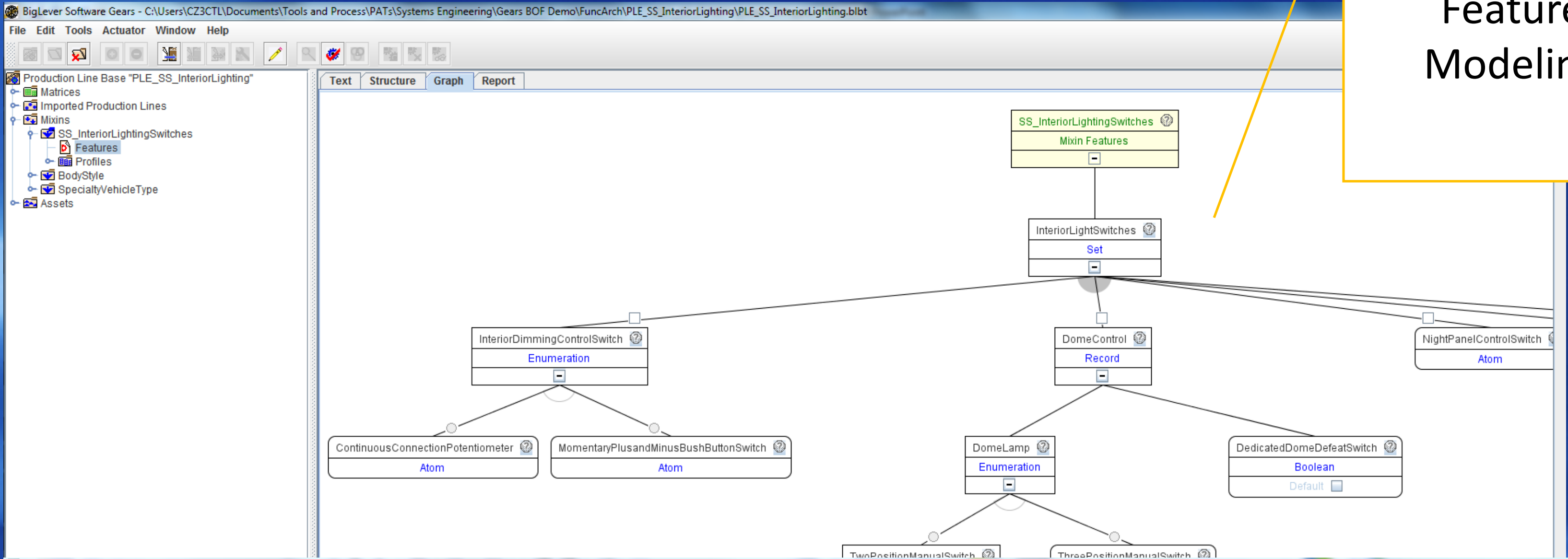
- Why Product Line over Products for GM Embedded Software?
  - As much as an 85% reduction in effort for a second (third, fourth, etc.) application
  - As much as a 70% reduction in field claims overall

E&SW PLE Approach

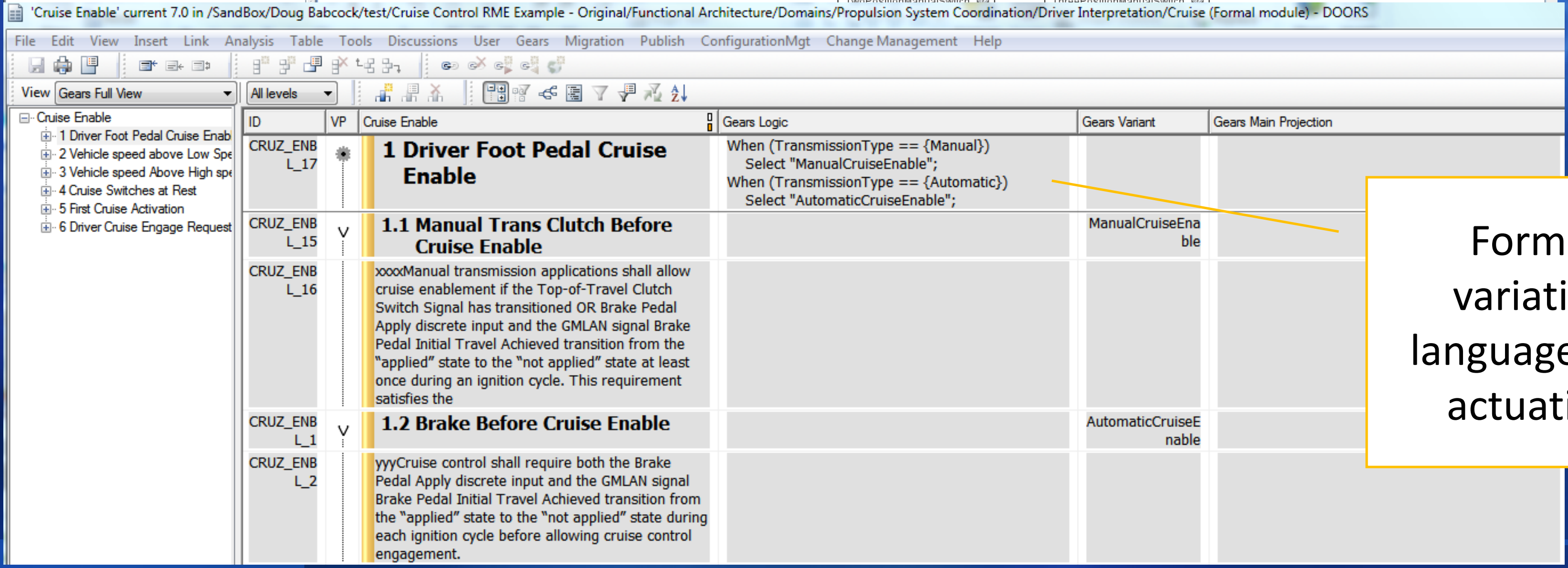


Feature Modeling Example

Rational Doors; Big Lever Gears in Combination


BigLever Software Gears - C:\Users\CZ3CTL\Documents\Tools and Process\PATs\Systems Engineering\Gears BOF Demo\FuncArch\PLE_SS_InteriorLighting\PLE_SS_InteriorLighting.blbt
File Edit Tools Actuator Window Help
Production Line Base "PLE_SS_InteriorLighting"
Matrices
Imported Production Lines
Mixins
SS_InteriorLightingSwitches
Features
Profiles
BodyStyle
SpecialtyVehicleType
Assets
Text
Structure
Graph
Report
SS_InteriorLightingSwitches
Mixin Features
InteriorLightSwitches
Set
InteriorDimmingControlSwitch
Enumeration
DomeControl
Record
NightPanelControlSwitch
Atom
ContinuousConnectionPotentiometer
Atom
MomentaryPlusandMinusBushButtonSwitch
Atom
DomeLamp
Enumeration
DedicatedDomeDefeatSwitch
Boolean
Default
TwoPositionManualSwitch
ThreePositionManualSwitch

Feature Modeling


'Cruise Enable' current 7.0 in /SandBox/Doug Babcock/test/Cruise Control RME Example - Original/Functional Architecture/Domains/Propulsion System Coordination/Driver Interpretation/Cruise (Formal module) - DOORS
File Edit View Insert Link Analysis Table Tools Discussions User Gears Migration Publish ConfigurationMgt Change Management Help
View Gears Full View
All levels
Cruise Enable
1 Driver Foot Pedal Cruise Enab
2 Vehicle speed above Low Spe
3 Vehicle speed Above High spe
4 Cruise Switches at Rest
5 First Cruise Activation
6 Driver Cruise Engage Request

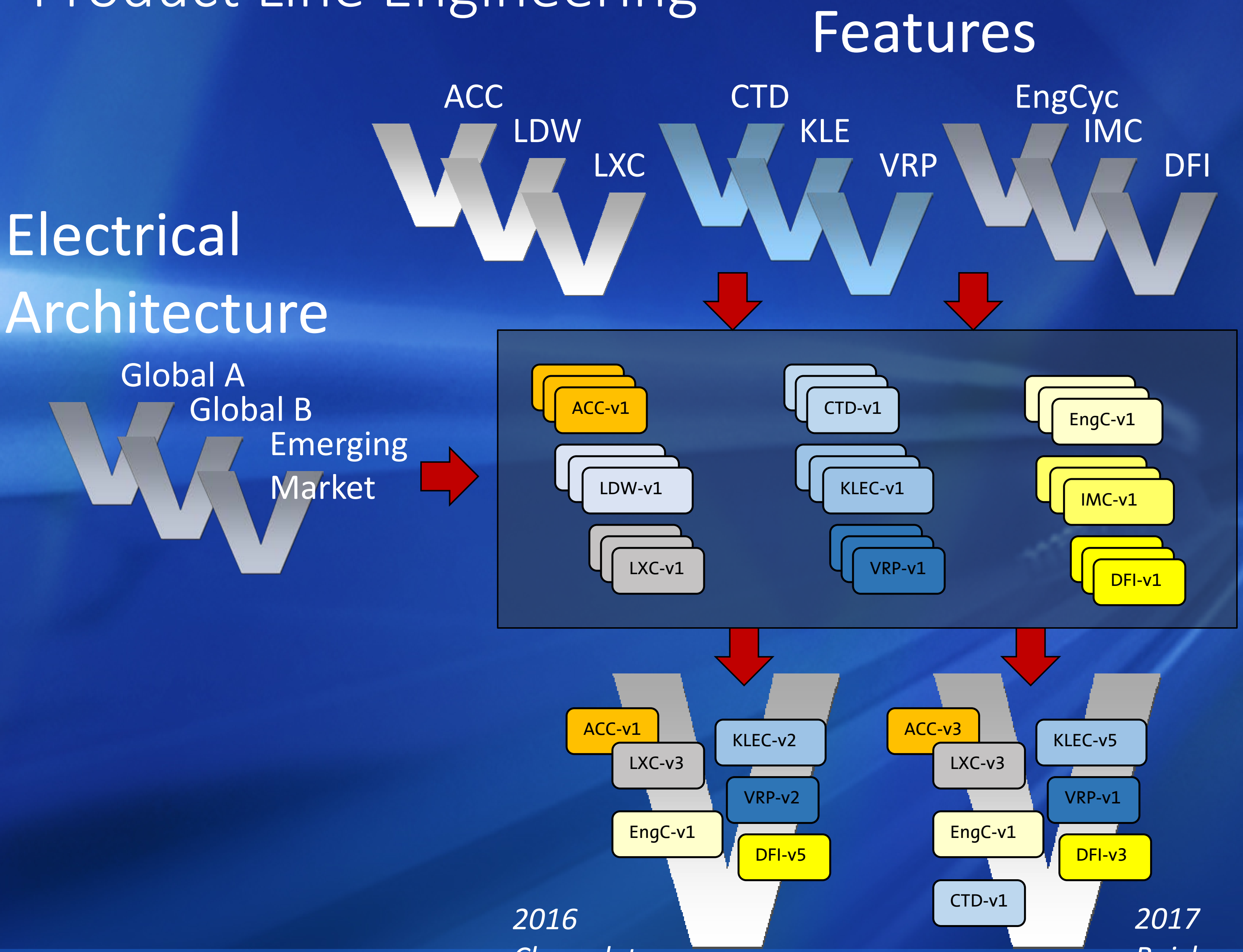
ID	VP	Cruise Enable	Gears Logic	Gears Variant	Gears Main Projection

CRUZ_ENBL_17	✹	**1 Driver Foot Pedal Cruise Enable**	When (TransmissionType == {Manual})
			Select "ManualCruiseEnable";
			When (TransmissionType == {Automatic})
			Select "AutomaticCruiseEnable";		

CRUZ_ENBL_15	V	**1.1 Manual Trans Clutch Before Cruise Enable**		ManualCruiseEnable	
CRUZ_ENBL_16		xxxxManual transmission applications shall allow cruise enablement if the Top-of-Travel Clutch Switch Signal has transitioned OR Brake Pedal Apply discrete input and the GMLAN signal Brake Pedal Initial Travel Achieved transition from the "applied" state to the "not applied" state at least once during an ignition cycle. This requirement satisfies the			
CRUZ_ENBL_1	V	**1.2 Brake Before Cruise Enable**		AutomaticCruiseEnable	
CRUZ_ENBL_2		yyyCruise control shall require both the Brake Pedal Apply discrete input and the GMLAN signal Brake Pedal Initial Travel Achieved transition from the "applied" state to the "not applied" state during each ignition cycle before allowing cruise control engagement.			

Formal variation language and actuation

Product Line Engineering



Features

ACC
LDW
LXC

CTD
KLE
VRP

EngCyc
IMC
DFI

Electrical Architecture

Global A
Global B
Emerging Market

ACC-v1
LDW-v1
LXC-v1

CTD-v1
KLEC-v1
VRP-v1

EngC-v1
IMC-v1
DFI-v1

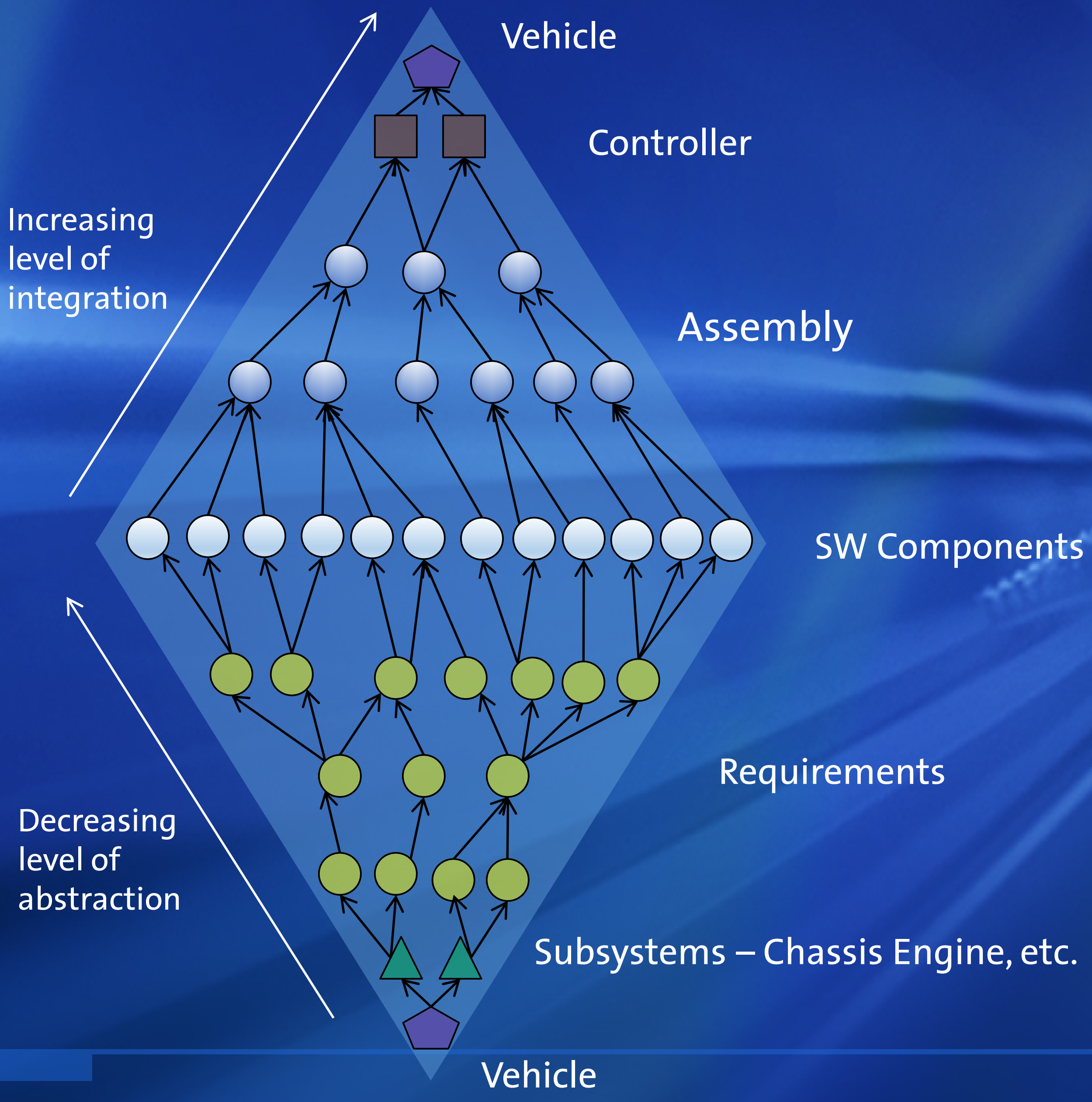
ACC-v1
LXC-v3
EngC-v1
KLEC-v2
VRP-v2
DFI-v5

2016 Chevrolet Volt

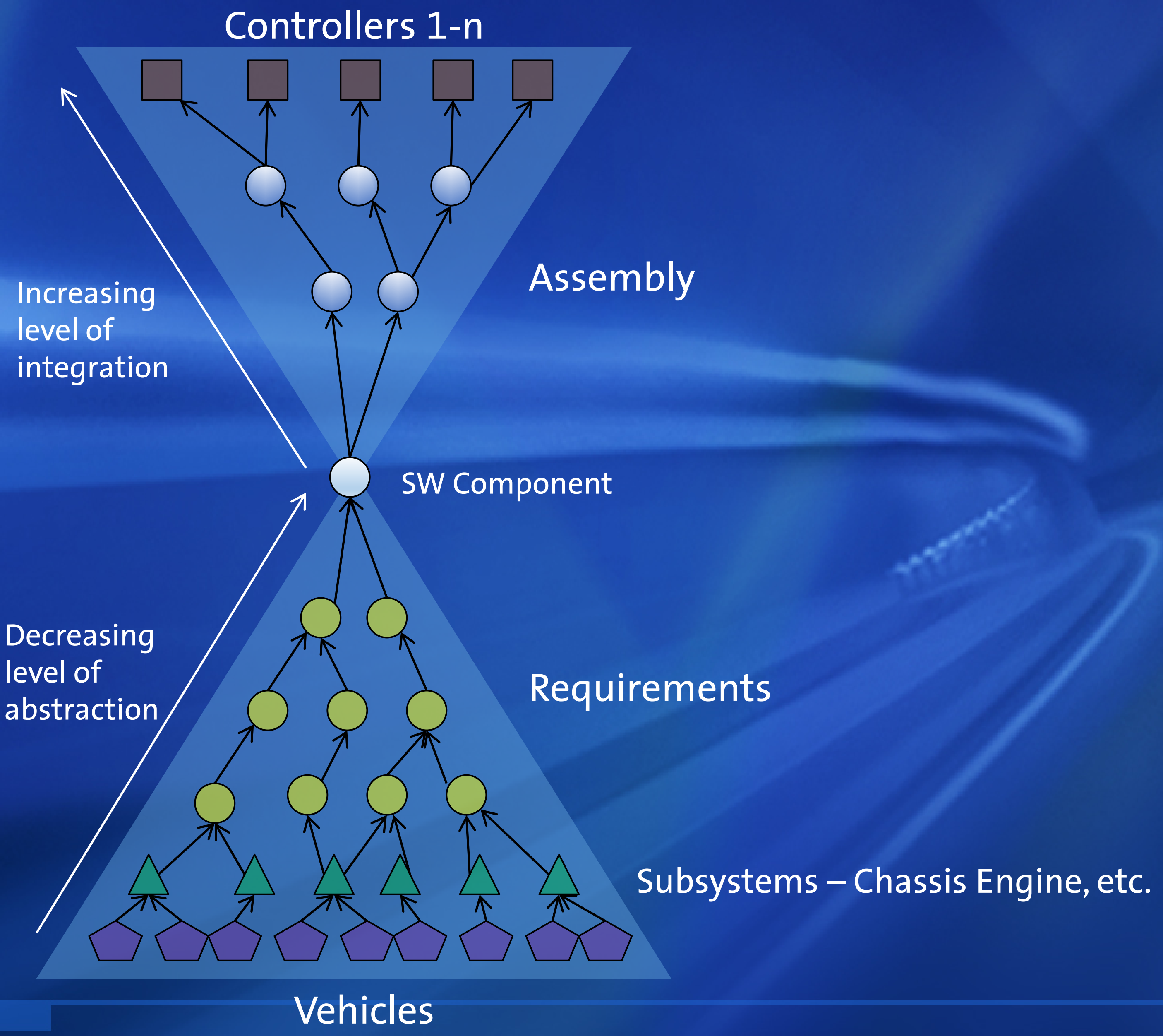
ACC-v3
LXC-v3
EngC-v1
CTD-v1
KLEC-v5
VRP-v1
DFI-v3

2017 Buick Regal

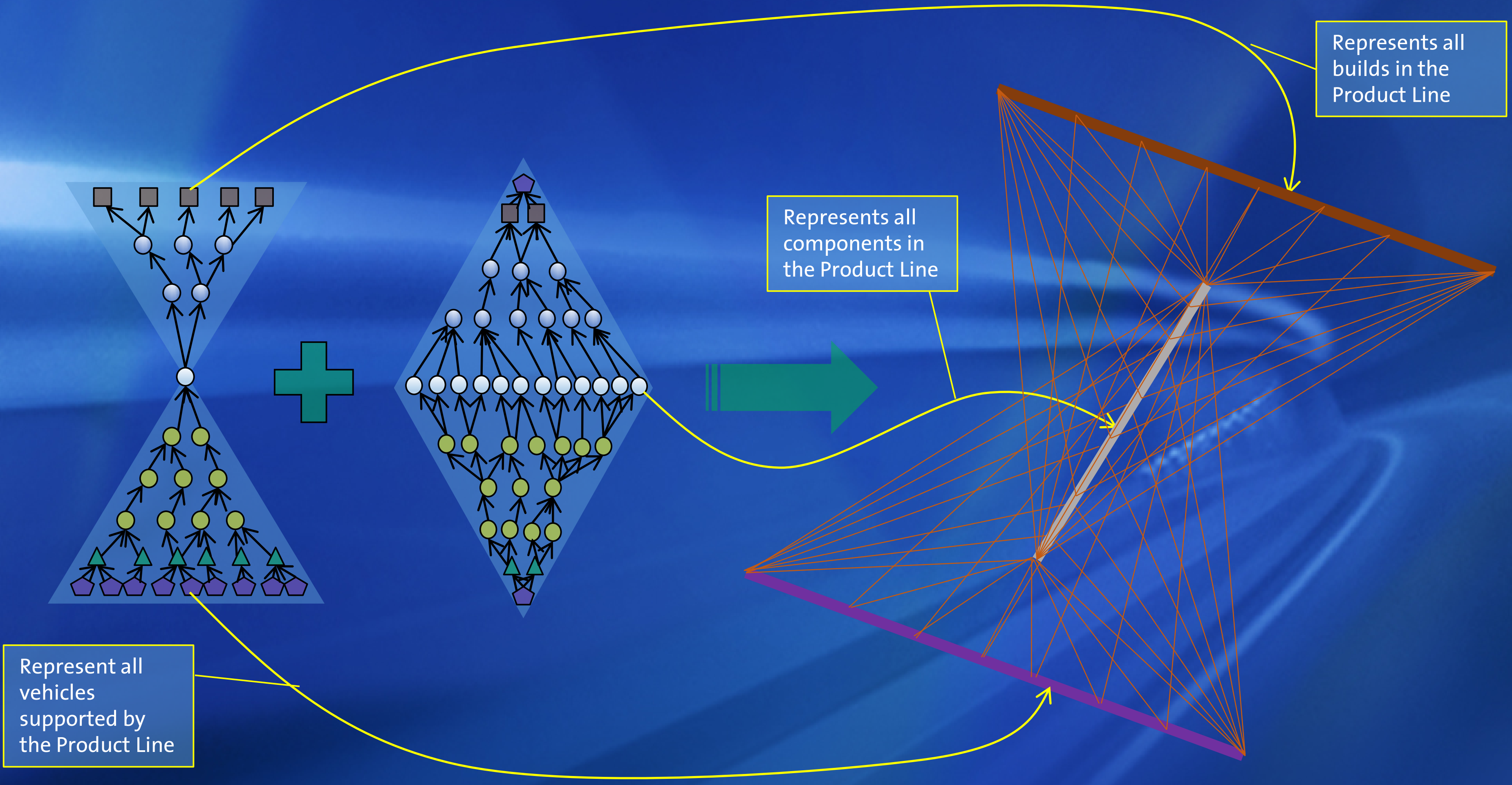
Software Product Line - Single Vehicle View


Vehicle
Controller
Increasing level of integration
Assembly
SW Components
Requirements
Decreasing level of abstraction
Subsystems – Chassis Engine, etc.
Vehicle

Software Product Line - Single Component View


Controllers 1-n
Increasing level of integration
Assembly
SW Component
Decreasing level of abstraction
Requirements
Subsystems – Chassis Engine, etc.
Vehicles

Software Product Line - Components X Vehicles

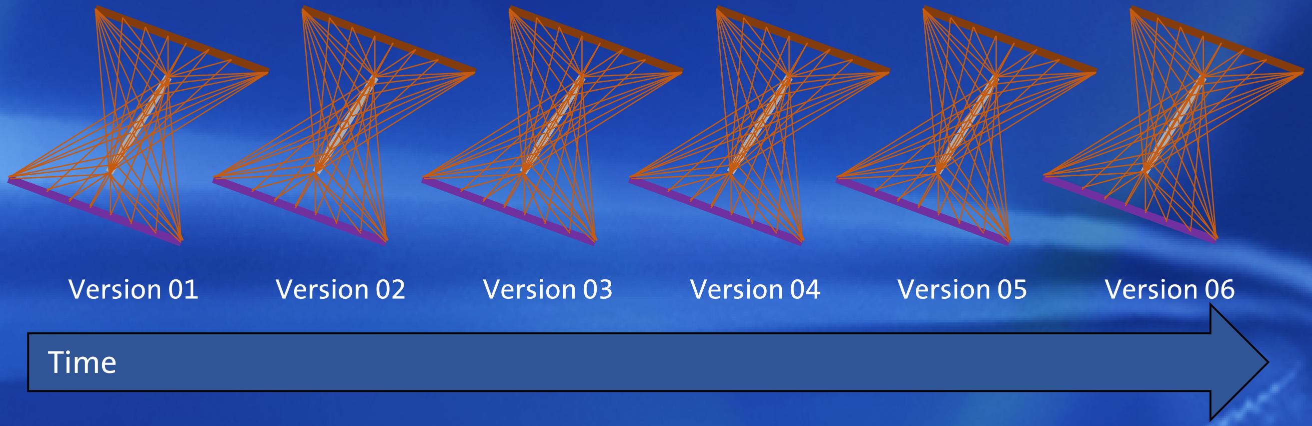


Represents all builds in the Product Line

Represents all components in the Product Line

Represent all vehicles supported by the Product Line

Software Product Line - One Stream



Version 01 Version 02 Version 03 Version 04 Version 05 Version 06

Time

- Only some objects change between instances, objects that do not change are inherited from previous instances
- Inheritance is **required** to be an automatic process handled by our tools, not a manual process
- GM Product Line has been a continuous stream since the late 1990s

HANDLE COMPLEXITY VIA PRODUCT LINES

- ❑ GM Releases A Coordinated Software Product Line Every 7 Weeks
  - Individual builds are guaranteed to work together
  - Multiple model years in parallel
  - Tuned for specific vehicle content & performance
  - Variation support and management is critical

- ❑ One Team, One System Design Tool Chain and One Process, Globally
  - Tools must migrate globally within a defined tool catalog
  - Older versions of tools must remain available as we enter vehicle production

Building The Plane As We Fly

- "General Motors has what is probably the largest product line (and the largest product line *engineering* problem/opportunity) on the planet.

  They are pushing into new PLE territory in almost all areas, from the sheer scale of their product line, to the size and structure and depth of the many engineering organizations who must cooperate to build it ...

  GM is years and years ahead of any other product-producing organization."

  Dr. Paul Clements
  Austin, TX – September, 2013

- Dr. Clements is Vice President of Customer Success at BigLever Software, Inc., where he works to spread the adoption of systems and software product line engineering.
- Prior to this, he was a senior member of the technical staff at Carnegie Mellon University's Software Engineering Institute, where for 17 years he worked leading or co-leading projects in software product line engineering and software architecture documentation and analysis.
- Co-wrote the SEI's first product line case study, was co-creator of the SEI Framework for Product Line Practice, was an author of the book *Software Product Lines: Practices and Patterns*

ECS Content & Development Process

- ECS Content
  - Software, ECUs, Gateways, Network Buses, S/As, CAN/LIN/Ethernet Controllers, Camera/Radar/Lidar Devices, Transreceivers,
- Unique aspects of the process
  - ECS Assets at every level developed as a productline
  - 3- tier approach to Development:
    - Tier 1: E/E Architecture (HW, S/A, Networks)
    - Tier 2: Software Architecture (Autosar)
    - Tier 3: Feature Development (Algorithm & Code)
  - Extensive `bench’ and `vehicle’ validation – HIL, PIL, VIL tests

SW to System Engineering

- Strong move to system engineering
- System level concerns like Safety, Security, Fault-tolerance, Diagnostics/Prognostics, assuming center place
- ISO 26262 based Functional Safety Concept
  - Template based Determination of Safety Goals
  - Various Early analysis like PHA, FTA, leading to safety requirements

Is there MBD?

- MBD approach successfully followed at feature level – Control Algorithm Design
  - Control Algorithms: Simulink / Stateflow, Rhapsody
- Plant modeling: Simulink, Saber, GT Power, AmeSim, CarSim, …
  - During Model-in-Loop and Software-in-Loop Validation
- Modeling Network Architecture has recently started
  - SymtaS Tool
- Electrical Wiring Modeling: Design Architect, Siemens NX
- Major effort in Product-line Modeling has been initiated
  - Gears based specification of variability in DOORs requirements
- Recently Rhapsody based System Modeling also initiated

Is there Sufficient MBD?

- Models mainly used for end artifact generation
  - Less used for Analysis: functional and Performance
- Subsystem and system level models are absent or limited (to structural level)
- Multiple levels of integration happens in the process but not sufficient application of MBD in integration

Is there Sufficient MBD?

- ECS Development appears isolated with apparent reference to physical/environmental element being observed/controlled
  - Physical/environmental elements considered at bench/vehicle level testing leading to late and costly design iterations
- No unified top-down view of System and Modeling that flow from requirements to models to system components and elements.
  - Traceability to Requirements exists but weak

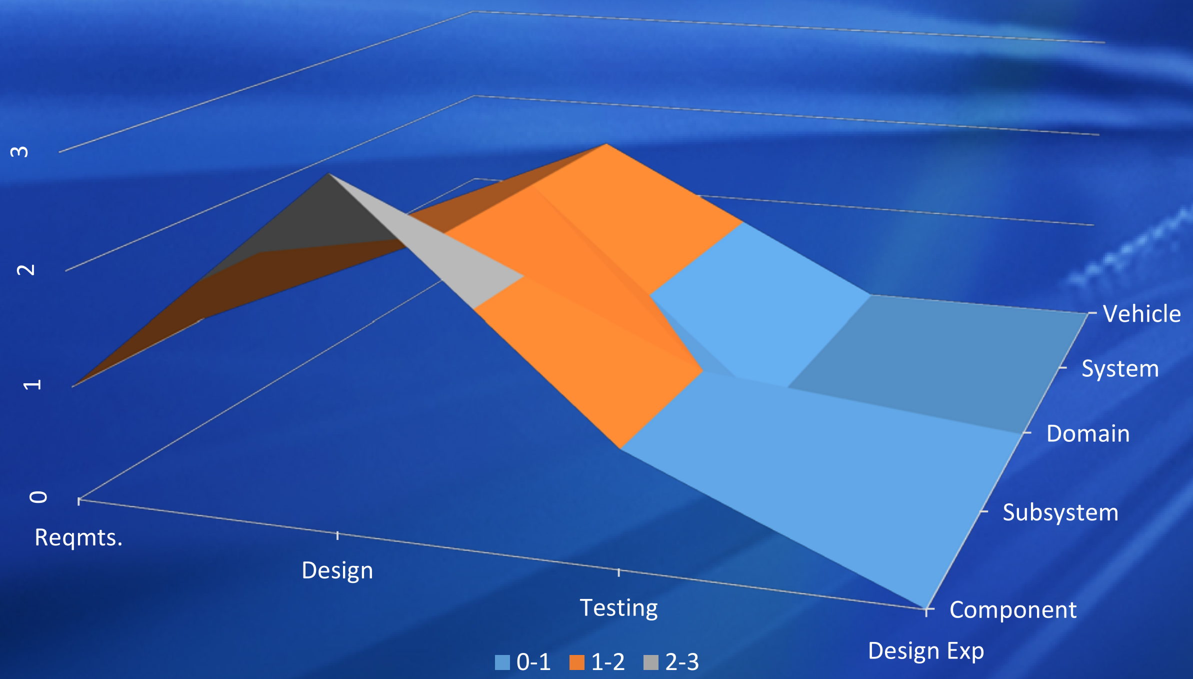
MBD at System Level

- End-to-end concerns and traceability
- Where is the end?
  - Human driver, passengers, other vehicles, drivers, infrastructure(local and remote) all could be involved
- Spatial Dimension is the most common view
- What about temporal dimension?
  - Build components or subsystems for not only today's requirements and conditions
  - What can change tomorrow? Physical components degrade, people change, people to automation, . . .

Is there sufficient MBD?

- Modeling Maturity
  - Measured along three dimensions
- Artifacts Complexity
  - Level 0: Human based
  - Level 1: Document based
  - Level 2: Structural Models
  - Level 3: Functional Models
  - Level 4: Performance Modeling and Design Space Exploration
- System Hierarchy
  - Components, Subsystems, Domains, Systems, Vehicle
- System Lifecycle
  - Requirements, Design, Testing and Validation

Chart Title


3
2
1
0
Reqmts.
Design
Testing
Design Exp
Vehicle
System
Domain
Subsystem
Component
0-1
1-2
2-3

Is MBD Sufficient?

- Model based Development vs Document vs People based Development
- Can models replace documents and people?
- Institutions and People
  - storehouses of knowledge, choices and decisions
  - Long history for traditional industries at least
  - Precedes the era of MBD
- Can today's modeling languages and tools capture these?
- Or should they?

Needed Enhancements

- Strong requirement driven approach to development
    - Requirements include assumptions on the physical/environmental elements
- Comprehensive Model based development
    - Models for SW (feature & Infra.), HW (platform, network, S/A), Physical (Plant under control) and Environmental elements (Road infrastructure, Human elements, other vehicles, etc.)
- Models at multiple levels of fidelity and composability of models at different levels of fidelity
- End to end traceability from traceability from requirements – models – reality, established through validation

Realization Challenges – Technical

- Availability of models from different disciplines
  - Adoption of MBD in every domain essential in the long run – advancing the eco system the same level
- Composability and Cosimulation of multi-domain models at varying levels of abstractions
  - Common semantic framework and interfaces
  - Very challenging across disciplines
- Efficiency of Co-simulation
- Tying up with requirements
  - Requirement engineering well developed for SW.
  - Extending to HW, Physical and environmental domains
- Product-line optimization & Correctness
- Availability of methodology and tools

Realization of Challenges – Organization

- Justification of significant additional work
  - Articulation of Value Proposition
- Efficient Auto Code Generators key reason for adoption of MBD in Control Algorithm Development
- How do we build on this to extend MBD to other areas: Integration, System Design, System Requirement, Analysis and Test Case Generation
- Best Practices, Process Standards would help
- Robust Tool Support Needed
  - SysML? Modelica? FMI? Autonomie?

The Baggage of Legacy and Industry

- Automotive E/E started long back (in the late 80s) even before MBD methods were usable in the industrial setting
- Large time-tested code base
  - Reverse Engineering
- Current production and process can not be stopped
- How do we introduce any new methodology seamlessly with the ongoing and existing process?
- ECS development carried out with large number of Suppliers building separate components and OEM combining all of them
  - New Set of suppliers coming in as Physical Component Model providers
- How do we have a common methodology across wide variety of suppliers?

Experimentation in Realization

- GM R&D has been working on this under `Lab to Road' Program
- Closely working with Engineering and other Industry partners
- Collaboration with various Universities
  - NECSIS - 10 Institutions
  - CMU, MIT, Vanderbilt U., UoM

Thank You!