Enhanced Data Dictionary
for model based automotive production software development

Todd Nordby, Technical Specialist
Agenda

- About
- Background and context
- Our organization’s workflow and motivation
- Solution
- Status
- Summary
About NAVISTAR

- **Major manufacturer** of commercial trucks, buses and defense vehicles
- **$11 billion** in revenue in 2019
- **Year-over-year market share increase** in the last five years
- **Customer-centric DNA** and industry-leading focus on customer uptime
- **Global alliance with TRATON Group** speeding technology innovation and further cost improvement
About Global Product Development

• Global Product Development (GPD) is an engineering organization within Navistar

• Controls & Software (C&S) is a GPD group responsible for designing and implementing engine and vehicle electrical and electronics components
  ▪ engine and vehicle control applications
  ▪ embedded software
  ▪ electronic component integration

• Model Based Design (MBD) within Controls & Software
  MBD used extensively for engine and vehicle control applications
  ▪ Algorithm development
  ▪ Production intent software code generation
Agenda

- About
- Background and context
  - Model Based Design (MBD) and Components
  - Data objects
- Our organization’s workflow and motivation
- Solution
- Status
- Summary
Component Based Development and Architecture

- Create components and make connections

Simulink top model with Reference Models

or AUTOSAR Composition with SWCs
Top-down vs. Bottom-Up workflows

- **Top-down**
  - Authoring defines components
  - Authoring makes connections
  - Create ARXML
  - Create skeleton models

- **Bottom-up**
  - Create models independently
  - Generate ARXML
  - Authoring integrates components
  - Authoring connects existing inputs/outputs
Data Management for MBD

- Data objects elaborate graphical designs
  - Define interfaces
  - Describe behavior
  - Control deployment/code generation

Where are these Simulink data objects stored?
- workspace (e.g. from .m or .mat files)
- Simulink Data Dictionary
Agenda

- About
- Background and context
  - Our organization’s workflow and motivation
    - Functional/algorithm focus vs software implementation
    - Agile architecture development
    - Production code generation
- Solution
- Status
- Summary
Functional focus

• Functional focus vs. software implementation

AUTOSAR.Signal => Rte_IRead_vwatp_Step_uAmbTRaw_uAmbTRaw
Organic architecture using bottom-up

- Can we facilitate “implicit authoring”?
  Ensure I/O Signal names exactly match

- Can we grow the system organically?
  Add components as appropriate
Streamlined production code generation

• Can we go from functionally focused MBD to production code easily?

How easily can we go from here... to here?
Agenda

- About
- Background and context
- Our organization’s workflow and motivation
- Solution
- Status
- Summary
Simulink Data Dictionary advantages

• Design data management
  • Create and manage data object definitions
  • Specify design data using Model Explorer interface

• Persistent repository
  • Repeated data loading not required
  • Automatically associate design data with model
Simulink Data Dictionary gaps

- Functional focused support (abstraction)
  - Uses data object class not abstract type
  - Includes code generation specifics

- Organic architecture (parallel development and implicit authoring)
  - Doesn’t match input/output names explicitly
  - Bottom-up approach possible but integration complicated
Navistar Enhanced Data Dictionary

• Goals
  • Enhance and add functionality
  • User friendly (particularly function developers)
  • Support Controls & Software organization’s workflow

• Approach
  • Simulink Data Dictionary as “database”
  • MATLAB Engine API for Java
Data abstraction

- Functional view
  - Inputs
  - Outputs
  - Calibrations
  - etc.
Implementation hiding

• Simulink guides data management (helpful)
• Simulink requires implementation specific knowledge (inconvenient)
• Algorithm developer vs. software engineer
Implementation hiding concept

- User enters functional attributes
  - Min, Max, Units, etc.
- Hide and automate deployment requirements
  - Object Class
  - CoderInfo
  - etc.

that was easy...
Implementation hiding details

- Hide and automate deployment requirements
- Directly support production code generation

I'm glad I don't need to know this...
Functional developer assistance

- Consolidated model synchronization
  All missing/extra data objects vs Simulink 1 type at a time
Functional developer assistance

• Typical use case automation
  • Automatically select ‘single’ datatype on creation (project preference)
  • Fixed-point datatypes automatically defined from slope/offset
  • Boolean datatype automatically sets Min/Max
  • Min/Max checked on data entry
Enable organic & parallel architecture development

- Enable and enforce compatibility
  - Select inputs from other outputs
- Support parallel bottom-up growth
  - Allow manual input creation
Implementation considerations

• Duplicate I/O data objects
  • streamlined/independent component development

• requires integration reconciliation after component implementation

• supports “dangling inputs” for resolution later

• Simulink Data Dictionary references
  • complete project and coordination required

• component integration completed at implementation

• “dangling inputs” require immediate external component modification
Current status and beyond

- Initial Simulink Data Dictionary based tool launched Q4-2019
- Initial improvements and support for R2020a added Q2-2020
- Ongoing user feedback improvements in progress

- Currently in use on a production intent project
  - Already supported several vehicle intent software releases

- ToDo
  - Incorporate usability feedback
  - Can we take advantage of more built-in Simulink capabilities?
    - Embedded Coder Dictionary
    - Code Mappings Editor
Summary

- Navistar’s Controls & Software group uses Model Based Development (MBD)
  - to facilitate embedded application development by 100’s of engineers
  - on 4 projects and growing
  - including 2 projects in production

- Navistar’s Enhanced Data Dictionary
  - Supports Controls & Software’s functional focused organization
  - Provides robust and streamlined component based workflow
  - Supports parallel component development
  - Eases production code generation
THANK YOU