Training Part 1
Overview

Sponsored by Lee Slezak
PSAT Used to Support R&D and Management Decisions

- After a thorough assessment, PSAT was selected in 2004 as the primary vehicle model for all FreedomCAR and 21 CTP activities by the U.S.DOE, stating that “All future code development and enhancements for OFCVT shall focus on PSAT and PSAT-PRO”
- PSAT was awarded a R&D100 Award in 2004 and a Technology Transfer Award in 2007
- PSAT is currently used by more than 130 companies and 700 users worldwide, including GM, Ford, Chrysler, Hyundai, Toyota
Reliance on Modeling & Simulation is Continuously Increasing

- Reduce cost and time to production
  - Provides math-based environment for more thorough multidisciplinary integration and testing in the virtual environment before hardware builds
  - Sorts technologies quickly to reduce hardware build iterations
  - Promotes parallel and integrated virtual development of control systems and hardware
  - Reduces/eliminates duplicate modeling and analysis work and activities

- Enables fast to market with new technologies and real fuel economy
  - Delivers better-integrated, initial designs that balance Fuel Economy, Emissions and Drivability (FEED) requirements.
  - Provides common methods and tools for comparing/evaluating technologies.
  - Facilitates efficient, seamless link from research to production to maximize reuse of work products and eliminate waste.
The objective is to accelerate the development and introduction of advanced technologies through a Plug&Play architecture that will be adopted by the entire industry and research community.
Accelerating the Development and Introduction of Advanced Technologies

- Why is Autonomie Unique?
- Develop/Evaluate Component/Control Technologies
- Develop/Evaluate Vehicle Architectures
# AUTONOMIE/PSAT Comparison

## Architecture

<table>
<thead>
<tr>
<th>Capability</th>
<th>PSAT</th>
<th>PSAT-PRO</th>
<th>Autonomie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plug &amp; Play Architecture</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Hierarchical Architecture Standards (Vehicle, syst...)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Model Reusability through System Experts (Concept to Production)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Establish Standard Interfaces (Industry-wide)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

## Features

<table>
<thead>
<tr>
<th>Capability</th>
<th>PSAT</th>
<th>PSAT-PRO</th>
<th>Autonomie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model/data Customization</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Powertrain Configuration Customization</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Select Appropriate Level of Modeling</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>GUI Customization (process, post-processing...)</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Database Management</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
## AUTONOMIE/PSAT Comparison

### Usage*

<table>
<thead>
<tr>
<th>Capability</th>
<th>PSAT</th>
<th>PSAT-PRO</th>
<th>Autonomie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluate Fuel Consumption Benefits (technology, size, powertrain configuration...)</td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Evaluate and Balance FEED in Simulation (Fuel Economy, Emissions &amp; Drivability)</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Develop Component Requirements</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Simulate Single Component</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Develop System/Subsystem Requirements</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Develop Vehicle Level Control</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Develop System/Subsystem/Component Control</td>
<td></td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Component-in-the-Loop</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Software-in-the-Loop, Hardware-in-the-Loop...</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

*Final usage depends on the level of details of the models available
Develop a Software Architecture & Environment to Plug-and-Play Math-Based Models of Hardware & Control System Software Early or Up-Front in the Design & Analysis Process

- Maximum User Access
- Maximum Flexibility
- Selectable Complexity
- Code Neutrality

Database

Interface

Models, Data

Graphical User Interface

Models, Data

- Setup Simulation
- Generic Processes
- Results Visualization
- Linkage with Other Tools

User Access Control

Enterprise Wide Solution

Version Control

Database Search

Database
Model & Data Requirements

- **Maximum Reusability**
  - Automated integration of existing models / controls / data
  - All models for a specific area of expertise in a single location
  - Systems duplicated using Matlab API

- **Maximum Flexibility**
  - Any system can be built automatically
  - User can add their own configurations
  - Single components or entire vehicles can be simulated

- **Selectable Complexity**
  - Common nomenclature (i.e., naming, I/O...)
  - Common model organization (i.e., CAPS)
  - Model compatibility checked

- **Code Neutrality**
  - Matlab / Simulink main environment
  - Use S-functions
  - Co-simulation (i.e., CoSimate*)

* Not yet implemented
Graphical User Interface Requirements

- **Setup Simulation**
  - Select architecture, model and data
  - Check compatibilities
  - Select simulation outputs to Workspace
  - Select simulation type (i.e., fuel efficiency, performance...)

- **Generic Processes**
  - Calibration, Validation, Tuning
  - Parametric study, including Monte Carlo analysis
  - Optimization algorithms
  - Predefined set of simulations & report (i.e., Drive Quality)

- **Results Visualization**
  - Predefined calculations (i.e., fuel economy, efficiency, power...)
  - Predefined plots, both time based and lookup tables
  - Energy balance
  - Specific plots available for different models (users defined)

- **Linkage with Other Tools**
  - Co-simulation (i.e., CoSimate*)
  - Specialty Tools (i.e, GT-Power, CarSim, AMESIM, AVL Drive*...)”
  - Database Management (i.e., SourceSafe...)
  - Well-to-Wheel (i.e., GREET*)

* Not yet implemented
### Database Requirements

<table>
<thead>
<tr>
<th>Category</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Access Control</td>
<td>- Prevent unauthorized users from accessing restricted or proprietary data</td>
</tr>
<tr>
<td></td>
<td>- Allow authorized users to download all necessary files</td>
</tr>
<tr>
<td></td>
<td>- Ensure model is documented before integration into database</td>
</tr>
<tr>
<td>Enterprise Wide Solution</td>
<td>- Allow users to collaborate (i.e. share models)</td>
</tr>
<tr>
<td></td>
<td>- Main database accessible anywhere</td>
</tr>
<tr>
<td></td>
<td>- Consistent process for interacting with files</td>
</tr>
<tr>
<td>Version Control</td>
<td>- Maintain traceability of all changes</td>
</tr>
<tr>
<td></td>
<td>- Keep linked files together through entire vehicle process (i.e. design, simulation and test)</td>
</tr>
<tr>
<td>Database Search</td>
<td>- Use keywords to search data, models, controls related to specific projects</td>
</tr>
<tr>
<td></td>
<td>- Quickly find the correct model with the correct fidelity of modeling and all related files</td>
</tr>
</tbody>
</table>

11
Key Benefits

- **Plug&Play**
  - Flexibility & Reusability
  - Customizable architectures
  - Common Nomenclature
  - Code Neutral

- **Reduces Cost & Time to Production**
  - Common Methods to sort technologies quickly to reduce hardware build iterations
  - Reduces/eliminates duplicate modeling and analysis work
  - Delivers designs that balance Fuel Economy, Emissions and Drivability (FEED) requirements

- **Enterprise Wide Solution**
  - Database Management
  - Provides common methods and tools for comparing/evaluating technologies

No other tool currently allows the linkage with any legacy code!
Unique Feature - Plug & Play

Implement any language
Automated process to import
legacy code (data, model,
control, process)...

Legacy Code

Plug & Play

Legacy
Processes

Specialty
Software
(COTS)

Database
Management

Version Control
Database Search

CarSim*
GTPower*
Amesim*
AVL Drive...

Calibration
Validation
Tuning
Drive Quality...

No other tool currently builds the model automatically!

* Already linked
Key New Feature - Software Customization

- Model organization
  - Single or multiple plants
  - Controller location

- Fuel economy
- Validation
- Drive quality
- Control...

- Proprietary Information

- Customization

- Processes

- Post-processing

- Calculations
  - Plots
  - Reports...

- Models
- Data
- Controls...
Integrate Legacy Code

Example: Integrate Legacy Simulink Models

Legacy Code

GUI Automatically Renames Variables, Creates Necessary Support Files

Ready for use in

AUTONOMIE
Model Plug&Play Capability Provided by “Wrapper”

First block automatically built to select the required input parameters, change units and data type

Legacy Plant

Last block automatically built to change units (SI) and data type before sending the information to a bus to make them available to other systems
Experts Develop Systems

1- What is a System in Autonomie Terminology?
Experts Develop Systems
2- Each System Configuration Can Be Customized

Each System is Optional

Controller
Actuator
Plant
Sensor

Electric Machine #1

Example: GM 2 Mode HEV Transmission Plant

Electric Machine #2

Gearbox

Any System can have Subsystems To Accurately Represent Hardware
Experts Systems Can Be Reused
3- How Can We Reuse it?

Expert #9

Vehicle

Expert #3

Battery

Expert #1

Controller

Expert #2

Plant

Expert #7

Chassis

Expert #6

Engine

Expert #4

Controller

Expert #5

Plant

Expert #8

Driveline
Autonomie Designed to Be Used For All Steps in the Development Process

- Build and compare large number of technology, powertrain, options
- Ensure simulation traceability, model compatibilities
- Easy selection & implementation of data, models, control or cycles
- Run batch mode + Distributed computing
- Analyze and compare test and simulation data
- Enables MIL, SIL, RCP, HIL, CIL
- Database Management
- Generic Processes
Accelerating the Development and Introduction of Advanced Technologies

- Why is Autonomie Unique?
- Develop/Evaluate Component/Control Technologies
- Develop/Evaluate Vehicle Architectures
Develop Plant Models with Different Level of Complexity

- Steady State Model
- Physical Model
- Mean Efficiency Model
- Highly Dynamic Model with Production Code

- Different models are necessary to study different phenomena
- Detailed models are required when technology cannot be tested or does not exist!
Link with Commercial Off the Shelf Tools (COTS)

#1 – Develop Model in Native Environment

#2 – Integrate Model in Simulink

#3 – Use Model in Autonomie

COTS
(1) can be used to generate maps
(2) can be run in the Simulink environment
(3) can be run in their own environment (co-simulation)

GT Power, CarSim have already been linked
Select the Appropriate Level of Modeling

Controllers

Plants

Engine Experts

Battery, Motor Experts

Transmission Experts

Chassis Experts

Vehicle Experts
Main Steps of Model Based Design

**Software-in-the-Loop**
- Algorithm or Controller Model
- Plant Model
- Compiled C-code S-Function
- Execution on Host Computer
  - Non Real Time
  - No I/O

**Hardware-in-the-Loop**
- Algorithm or Controller Model
- Plant Model
- Embedded Target
- PC with I/O Boards

**Rapid Control Prototyping**
- Algorithm or Controller Model
- Plant Model
- Code Generation
- I/O
- Plant / Prototype

**Component-in-the-Loop**
- Algorithm or Controller Model
- Plant Model
- Code Generation
- I/O
- Rest of Vehicle is Emulated
- Entire System is Hardware
Develop Control Algorithm
Software-in-the-Loop

- Specific configuration defined to develop and test new control algorithms

Real Time Operating System (RTOS) ensures call of functions at specific intervals (such as CAN)

New algorithm(s) to be tested

Production Code

Hardware input/output

Sends and receives CAN signals
Test Low Level Control Algorithm
Hardware-in-the-Loop and Software-in-the-Loop

Hardware-in-the-Loop
(Control Hardware / Emulated Plant)

Automatic building save days of development

All the lines connecting the blocks are built automatically

Block built automatically
- Change units
- Change data type
- Extract all the proper sensors
Evaluate Fuel Consumption Benefits
Model-in-the-Loop

Example “Evaluation of HCCI Engine Fuel Savings for Various Powertrain”

Engine maps from UofM

Vehicle & Control Development

Fuel Consumption Analysis

Detailed Analysis of Reasons Behind Benefits
Evaluate Non Modeled Phenomena With Hardware Component-in-the-Loop

Example #1: Impact of battery cold start on PHEVs Fuel Consumption

Example #2: Impact of emission and engine cold start on PHEVs Fuel Consumption

Example #3: Engine and Battery are Coupled

Rest of the Vehicle Modeled

Battery behaves as if in vehicle

Engine behaves as if in vehicle

Commands
(Battery, ABC150, equipment...)

Sensors

Commands
(ICE, Dyno, equipment...)

Sensors
Evaluate Non Modeled Phenomena With Hardware Component-in-the-Loop - Examples

**Impact of Vehicle Level Control on Battery Life**

- Normalized # deep discharge cycles before EOL
- Depth of discharge measured
- Engine CO2 emissions (g/sec)
- Battery life (cycles)

**Impact of Vehicle Level Control on Engine Emissions**

- Vehicle speed (meters per second)
- Engine torque (measured)
- Engine emissions (CO2 g/sec)

**Impact of Temperature on All Electric Range**

- Initial temperature: 20 degrees C
- Initial temperature: 0 degrees C
- Initial temperature: -7 degrees C
- State of charge, vehicle speed, time in seconds

**Impact of Charger Rating on Battery Roundtrip Efficiency**

- Battery roundtrip efficiency (%)
- Battery charging current (A RMS)
- Charger power (kW)
Define Component Requirements

Component Data from R&D Teams

Vehicle Simulations

Vehicle Requirements

Sizing & Simulation

DOE / USABC Requirements

Battery RCP

Vehicle Testing

JCS VL41M

Battery

Electric Machine

Accessories

Testing

Model

Vehicle Classes

2004 Prius

Camry A/C Power

Prius A/C Power

Battery

Electric Machine

Accessories
Accelerating the Development and Introduction of Advanced Technologies

- Why is Autonomie Unique?
- Develop/Evaluate Component/Control Technologies
- Develop/Evaluate Vehicle Architectures
Validate Vehicle Models to Ensure Studies Integrity

Collect Test Data

Analyze Data

Develop Control

Validate Model

Engine Torque

Understand Gearbox

Model Gearbox

Source: GM

Example of GM 2 Mode Tahoe
Develop Vehicle Control to Maximize Fuel Displacement

**Global Optimization**

\[
\begin{align*}
T_{\text{eng}} & \rightarrow \omega_{\text{eng}} \downarrow \\
+ & \\
T_{\text{mc}} & \rightarrow \omega_{\text{mc}} \\
\end{align*}
\]

**Backward model**

\[
\begin{align*}
t & \rightarrow \{p_{n}(t), \text{SOC}(t), J_{w}(t)\} \\
\end{align*}
\]

**Bellman Principle**

\[
\begin{align*}
\chi(0) & \leftarrow \chi(t) \rightarrow \{p_{n}(t), \text{SOC}(t), J_{w}(t)\} \\
\end{align*}
\]

\[
\begin{align*}
\text{Optimal Control,} \\
\text{Minimal Fuel Cons.}
\end{align*}
\]

**Control Design**

\[
\begin{align*}
\text{Various Control} \\
\text{Principles}
\end{align*}
\]

**Rule-Based Control Design**

\[
\begin{align*}
\text{Various Control} \\
\text{Strategies}
\end{align*}
\]

**Heuristic Optimization**

\[
\begin{align*}
\text{Existing Control Logic} \\
\text{DIRECT Algorithm}
\end{align*}
\]

**Optimally tuned Parameters**
Develop Vehicle Control Taking into Account Non Modeled Parameters

Example: Impact of Vehicle Level Control on Engine Emissions for PHEVs (collaboration with ORNL)

Step #1 – Evaluate Different Controls in Simulation

Step #2 – Verify the Trends with Engine-in-the-Loop

Evaluation of Different Vehicle Control Logic and Tuning

analyze results
Match Technologies, Configurations to Specific Applications

Impact of Aerodynamics for Different Line Haul Applications

Comparison of Aerodynamic Fuel Savings for Drive Cycles vs Steady States

Impact of Mild and Full HEV for Line Haul Applications

The Sum of the Combined Technologies < The Sum of Each Technology

Impact of All Technologies on Fuel Consumption

Class 2B Pickup

- Engine
- Transmission
- RR
- Cd
- Weight
- Hybrid improvements
- Baseline improvements
Evaluate Uncertainties Through Monte Carlo Analysis

- Uncertainty is modeled by a probability density function (pdf)
- How is the uncertainty propagated?

- PHEV 10 miles All Electric Range (AER) midsize used as reference case

Inputs: Cd, FA, Crr, Weight

Sampling: Monte Carlo (MC), Latin hypercube (LHS), Median Latin hypercube (MLHS), Quasi Monte-Carlo

Results:
A software environment and standard framework

Summary

- Unifies the entire engineering organization for efficient operation
- Focuses analysis on total system engineering and integration
- Provides complete user customization by an open architecture
- Simulates from single components, subsystems to entire vehicles
- Manages models, data, processes, results and control code from research to production by configuration and database management