Modeling with CANoe
by Bruce D. Emaus
President of Vector CANtech
& SAE Embedded Software Standards Committee Chair

Automotive Software Workshop – San Diego
March 2006

This is a Vector commercial...

Vector – The Company Bruce Emaus Works For

- Vector is a global company located in Stuttgart, Germany
  - Subsidiaries in USA, Japan, France, Sweden

- Market leader - Small Area Network tools – CANalyzer, CANoe, and CANape
  - Protocols include: CAN, LIN, MOST, FlexRay, ...

- Market leader - In-vehicle Embedded Software Components
  - Software stacks include: Ford FNOS, GM LAN, DC NET, and others
The auto industry uses the model for incremental integration

Modeling steps in the distributed product development process

- Partition into modules
- Design all modules (everything is virtual)
- Evaluate network traffic
- Transfer model to all involved groups
  - Each module development activity
  - Each module testing activity
  - Each vehicle integration testing activity
- Each individual module development activity uses the model
  - To evaluate the module with the remaining system
  - To add their own internal behavior model – if desired
  - To add their own special requirements to the model – if desired

Continues on next page...

... continued from previous slide

- Each individual module testing activity uses the model
  - To test the module with the remaining system
  - To add their own module-specific test requirements to the model
- Each vehicle testing activity uses the model
  - To test the complete product or portions of the product with the remaining system model
  - To add their own vehicle-specific test requirements to the model
  - To add their own special requirements to the model – if desired
- Vehicle troubleshooting activity uses the model
  - To debug product problems with the remaining system model
Today’s automotive business requirements for modeling

Use the model...

- To design the distributed product
- As a real-time executable specification
- To simulate and evaluate the overall distributed product design whether one network or more and whether one protocol or more
- To test the entire product or any portion of the distributed product
- To show and animate in real-time any portion of the product’s human interfaces – buttons, front panels, and indicators
- To show, animate, and activate product inputs and outputs
- To allow the supplier to accomplish their module development from design thru testing and especially allow the supplier to add their own models whether by adding or insertion of existing models (like Simulink)
- To start the next distributed product design – allows high re-use

Observation: The first deployment of this enterprise-wide modeling method and process consumes the most effort in comparison to re-using it a second time.

Automotive modeling requires visual information engineering
Modeling Across the Automotive Development Cycle

**Business value across all phases of the development**

- **Idea**
  - System Requirements
  - ECU Design
  - SW Modules & HW Modules

- **System Architecture, Sim.-Model**
- **Real ECUs + remaining-bus-simulation**
- **ECU (SW on HW) + remaining-bus-simulation**

- **Legends**:
  - Phase identified by results
  - Testing activities

- **Production**
  - EOL tests, ...
  - Use case tests, ...

- **Accepted System**
  - System tests "black-box" HIL-Tests, ...
  - "White-box" tests SIL-Tests, ...

- **OEM, Module Supplier**
  - "Black-box" tests, integration tests, functional tests, hazard tests HIL-Tests, EOL Tests, ...

---

Automotive Distributed Product Testing using Modeling

- **Flexible testing environment**
  - Completely simulated
  - Some modules simulated and others available as real modules
  - All real modules

- **Test execution tool**
  - Runs on the PC
  - All CANoe analysis features and windows (graph, trace, data, ...) are available.
  - Create your own test reports
ECU Testing with the model – using CANoe

- osCAN-library
- ECU Software Component
- ECU

- Debugging
- Memory check
- Coverage check
- Profiling

- Test

- real module
- simulated module
- partial real system
- remaining bus simulation

Automotive Distributed Product Testing with CANoe

- Other application (ex. Simulink)

- CANoe

- CAN
- LIN
- MOST
- FlexRay
- Diagnostic I/F
- KWP 2000

- Digital I/O
- Analog I/O

- GUI
- COM
- Panels

- DBC
- LDF
- XML
- CDD

Input files including...
Product message database

System under Test

Automotive Software Workshop – San Diego - March 2006
Black Box Testing with CANoe

- **Stimulation**
  - Already available constructs to output messages
  - Constructs to set single signals

- **Wait for event conditions**
  - Wait for message, key, environment variable, ... 
  - Wait for combinations

- **Evaluation and reporting**
  - Retrieve message and signal data
  - Use any CAPL function

```
// Stimulate
output(Req2MsgBuf);
SetSignal(Req1Msg::SigX, 5)

// Wait
TestWaitForMessage(RespMsg, 2000)

// Evaluate + report
TestValidateSignalInRange(RespMsg::SigA, 0, 1)
GetWaitEventMsgDataCAN(RespMsgBuf)
If(RespMsgBuf.Time > lMaxTime) {
  TestStepFail("Time", "Too late");
}
```

Test Programming in CAPL

Starting with an XML-based test specification

- **Stimulation parameters**
  - Input vector: Values, e.g. signal, environment variable

- **Evaluation parameters**
  - Output vector: values
  - Timing
    - Evaluation is done automatically

- **Control flow**
  - Is built-in to predefined ‘test patterns’, several patterns will be available
  - Not needed on specification level
    - Maintainability increased

- **Report**
  - Is written automatically
  - Can be formatted by changing the style sheet

```
<statecheck wait="2000">
  <input>
    <cansignal name="Ignition">1</cansignal>
    <cansignal name="Level" message="Wipers">2</cansignal>
  </input>
  <expected>
    <envvar name="WiperMotor"><gt>0</gt></envvar>
  </expected>
</statechange>
```

Black-Box-Test Specification in XML (Example)
Starting with an XML-based test specification - example

Example test specification in Altova XMLSpy

Alternate Testing Methods used with CANoe

Example Test Specification in Altova Authentic
Invariant Testing with CANoe

- Observing ‘invariant’ tests
  - Stimulation is done by e.g. a driving cycle
  - Mapping-functions are checked dependent to the state

- Definition of mapping-functions for all states of the System Under test is a challenge

- Additional criteria are needed to define the end of the test

**Blackbox Testing Capabilities (Theory)**

```
\[\text{Inputs} \quad \text{system or unit under test} \quad \text{Outputs} \]
\[\text{Input}_i = f_i(t) \quad \text{Output}_j = g_j(t, I, \ldots, J)\]
```

**Tester**

```
\text{Stimulus} \quad \text{Stimulus} \quad \text{Stimulus} \quad \text{Stimulus} \quad \text{Response}
```

**Unit Under Test**

**Creation of test specification based on test patterns**

- **Domain Knowledge**
  - CANdb++
  - CANdela Studio
  - DaVinci
  - DOORS
  - UML Tool
  - Excel
  - ...

- **Pattern-based Test Specification (XML)**

- **Legend:**
  - Test Case Generator
  - Test Case Generator

- **Legend:**
  - G

  - Example:
    - e.g. Check cyclic messages occurrences
    - e.g. Test all provided diagnostic services
    - e.g. Test mapping of input/output signals, timing conditions
    - e.g. Use sequence- and/or state-diagrams
    - e.g. Gateway routing table

- **Engineer**

- **Test Case Editor**
Simulation & Test Execution Environment

CANoe

Simulation of Node A

Model of Node (CAPL, Simulink)

CANoe Interaction Layer

Bus Signal Server

CAPL Signal Driver

Diag Server

I/O Port Server

Test Execution

CAN, LIN, MOST, FlexRay

I/O, Diagnostic Services

CANoe

System under test

Testing with CANoe - Future Perspective

- Further solutions and enhancements are planned
  - Extension of the Test Service Library (definition of invariants)
  - Extension in amount and functionality of ‘test patterns’
  - Test case generators for dedicated sources

- Testing with CANoe may be customized as project work
  - Specialized test patterns may be developed
  - Specialized test case generators may be developed
  - Specialized test report formats (HTML, ...) may be developed

- Interoperability to process tools may be customized
  - Dedicated process tool can be integrated as project work
  - Dedicated workflows can be provided (when using eASEE)
Can models be used as a real-time executable specification

- Using today’s typical fast PC, CANoe is capable of executing all automotive functions – with the exception of a few high speed motion control processes like powertrain control
- Example – a recent OEM vehicle with 2 CAN buses; one with 25 modules and the other with 15 modules – all simulated in real time

Note: Vector is producing a new Real-Time CANoe product to allow high speed motion control process models to be useable for development and testing in real time

- For the auto industry – the use of a real-time executable specification is gaining more interest

What about autocode generation from a model

- Autocode generation from a model is really a module specific activity and not at the level of the system (or the distributed product level)
- Still in its infancy with up to now a wide variety of results in the auto industry (the spectrum traverses good to bad), many (especially in management) are waiting for this new method of creating software to produce consistent positive results

Observation: Several in the auto industry think another ten years will be needed before the majority of suppliers will be using this model-based code generation technology as their primary way of doing module development.

- For the auto industry - Module-based modeling can be used in combination with the distributed product modeling
- HOW? Using CANoe, the module developer can insert their module-specific model (e.g. Simulink) into the distributed product model.
The basic automotive modeling process:
- The OEM makes a distributed product model – this is vehicle specific
- The OEM distributes the model to all "sourced" suppliers
- All suppliers create their individual modules
- Each supplier verifies the module behavior using the vehicle specific model
- All suppliers send their individual modules to the OEM
- The OEM integrates the modules onto the vehicle platform – bench-level and/or vehicle level
- The OEM tests, fixes, and completes the product

What tool is used by many automotive OEMs to accomplish this?
Vector’s CANoe

Applicability/main focus of Vector Tools
- DA Vinci or Simulink native
- Test Data Management
- CANoe
- CANstress
- CANscope
- CANister
Contacts

- Presenter
  - Bruce Emaus: bruce.emaus@vector-cantech.com
- CANoe Tool Manager
  - Rick Lotoczky: rick.lotoczky@vector-cantech.com
- Sales
  - sales@vector-cantech.com