ASCET in ECU Development Tool Chains
From Function to Software Development

- ASCET-MD/-RP
- ASCET-SE
- EHOOKS
- INTECRIO

RTA Tools

Meas. & Calibr. INCA
Meas. & Calibr. INCA-EIP
HIL Testing Labcar
FIL Testing

Virtual, & Rapid Prototyping

Modeling & Simulation

Software Model Design Tool
Automatic Codegeneration
Basic Software Modules

Realtime Operation

Rapid Prototyping

Meas. & Calibr. INCA-EIP
ASCET in ECU Development Tool Chains
Scope

- ASCET is the Tool for Automotive Software Development
  - Efficient creation of reusable, well-defined models and SW components
  - Easy validation of models through PC simulation and rapid prototyping
  - Fast and reliable code generation of high quality production code
  - Open interfaces for smooth integration into customer tool chains
ASCET in ECU Development Tool Chains
ETAS Experience in Production ECU Code

- Over 12 years experience in production code generation
  - First ECU series project using ASCET code generation has been 1998

- Number of cars with ECU software developed using ASCET:
  more than 100 million

- Number of ECUs running on ETAS OS & RTE:
  more than 400 million
ASCET in ECU Development Tool Chains
ASCET References include OEMs as well as Suppliers

**OEMs**
- BMW
- Daimler
- Volvo Truck
- Volkswagen
- Audi
- Isuzu
- NuCellSys
- GM
- Honda
- Peugeot
- Renault

**Supplier**
- **Bosch**
  - Chassis, Powertrain, Fuel Cell, Hybrid
- **Continental**
  - Body
- **Knorr-Bremse**
  - EBS for trucks
  - Brake system in ICE train
- **Getrag**
- **GM Allison**
- **Ricardo**
- **ZF Getriebe**
ASCET in ECU Development Tool Chains
Customer Projects Cover all Domains

• ASCET is the standard model-based development tool at BOSCH for
  • Powertrain (Diesel, Gasoline, Transmission, Hybrid, ...)
  • Chassis Systems Control (ABS, ESP, ...)

• Further Application Examples
  • AFS - Active Front Steering
  • Engine & Transmission Control
  • Fuel Cell Control
  • Body applications
    (e.g. seat adjustment, car access, rear-electronics-module)
  • Train Brake System
ASCET in ECU Development Tool Chains
Recent Customer Project

- Full Parallel Hybrid Drive in Porsche Cayenne & Volkswagen Touareg
- Software instead of mechanical solution for torque distribution between combustion and electrical engines
- Control software has been modeled and generated using ASCET
ASCET in ECU Development Tool Chains

Outline

- **Modeling with ASCET**
  - Control Models, Plant Models
  - AUTOSAR Support

- ECU Code Generation
  - From an abstract function model to ECU production code
  - Generate a complete ECU software

- Tool Chain Integration
  - What is needed beyond modeling and code generation
  - How does ASCET integrate in tool chains

- Conclusion
Modeling with ASCET
Graphical and Textual Models

- ASCET has well-defined & deterministic semantics
- Allows explicit control of execution sequence
- ESDL and Block Diagrams follow the same abstraction and can be used interchangeably

ESDL
Block Diagrams
State Machines
Modeling with ASCET
Table-based Models & Environment Models

Boolean Tables

Conditional Tables

Environment Models
- Continuous time / differential equations
- Specification of physical plant models
- Basic blocks can be defined in C or ESDL
- Graphical combination to model complex physical systems up to complete vehicle
- Variety of solvers available
Modeling with ASCET
Native Modeling of AUTOSAR SW Components

- ASCET supports AUTOSAR releases R2.1/R3.x
  - Modelling of AUTOSAR Software Components with native AUTOSAR model elements
  - Import of AUTOSAR Software Component Template (ARXML)
  - Generation of AUTOSAR compliant C-Code and corresponding ARXML
- Easy migration of existing models to AUTOSAR
Modeling with ASCET Library Support

- Two libraries with basic elements are provided with ASCET
  - Well-known ETAS SystemLib
  - Standardized MBFS* Library

- Contained elements include
  - Bit operations
  - Comparison
  - Basic transfer functions
e.g. PID, DT1, PT1

- Direct library access from specification editor
  - Fully configurable (e.g. frequently used)

*Model Based Function Specification v1.0 – ASAM AE
Modeling with ASCET
Customer Example: Modeling Single Functions

- BMW Modellotheke (model library)
  - Library of ASCET models for body functions
    - More than 15 functions have been modeled
    - AUTOSAR-compliant function models
  - Models are reused for many series and suppliers
  - Supplier receives model or generated C-Code for integration
Modeling with ASCET
One Integration Model for the complete ECU SW

- Operating System Configuration
  - Configuration of OS
  - Create tasks
  - Define task properties
  - Map functions to OS tasks & define their execution order

- C Code Integration
  - Integrated C Code editor
  - Allows interfacing to other software
    - allows anything C provides
  - Integrate functions with platform software (e.g. CAN Driver)

- Modeling functions and ECU integration – All in one tool...
BOSCH chassis systems uses ASCET to model
- Complete controller algorithm for ESP (Electronic Stability Program, including Anti-lock Bracking)
- Vehicle Dynamics Model (plant model)
- ASCET model is „single source“ for complete development methodology
- Only one model for PC simulation, rapid prototyping, tests, and ECU code generation
- System-wide resource optimization (RAM, ROM, run-time) for ECU SW generation possible

Klaus Meder, Vice President Engineering, BOSCH Chassis Control:
„ASCET is the development backbone for our entire software engineering process.“
„ETAS is a cooperative and reliable tool supplier. “
ASCET in ECU Development Tool Chains

Outline

- Modeling with ASCET
  - Control Models, Plant Models
  - AUTOSAR Support

- ECU Code Generation
  - From an abstract function model to ECU production code
  - Generate a complete ECU software

- Tool Chain Integration
  - What is needed beyond modeling and code generation
  - How does ASCET integrate in tool chains

- Conclusion
ECU Code Generation with ASCET
Motivation

- Requirements to ECU Software
  - Efficient in terms of memory and run-time
    - Non-optimal production code is a waste of ECU resources and money!
  - Safe and reliable
    - Value overflows or division-by-zero exceptions may lead to failure of the ECU!
  - Compliant to standards
    - Safety norms (MISRA-C, ISO 26262) or code compatibility (AUTOSAR)

- ASCET helps you to fulfill these requirements!
ECU Code Generation with ASCET
ECU Software uses fixed point arithmetic

- Physical values in function models are mostly continuous
  - E.g. temperature, torque, speed
  - Although most of today's micro-controllers have floating point units, these continuous values are coded as fixed point values in ECU SW
    - Fixed-point values consume less RAM
    - Fixed-point arithmetic has controlled numerical precision
    - Trade-off of precision vs. memory size is possible
  - Conversion to fixed-point is a mapping of physical values to integer values and is called „implementation“

![Diagram showing physical values and integer values with quantization and resolution.](image)
ECU Code Generation with ASCET
Fixed-Point Software Consumes Less RAM

- Assume a temperature sensor with range -50..+100°C
  - Sensor has 1/128°C (≈0.0078°C) precision
  - We use 100 variables in the code

- Single precision floating point
  - 100 x 4 bytes = 400 bytes

- Fixed point in 1 byte
  - 100 x 1 byte = 100 bytes
  - Less precision => 1°C resolution
  - 75% less RAM

- Fixed point in 2 bytes
  - 100 x 2 byte = 200 bytes
  - Required precision
  - 50% less RAM

- Could still save 50% RAM even if the sensor had twice the resolution
  - 1/256°C (≈0.0039°C)
  - Using an offset, so will be slower
ASCET supports:
- Definition of conversion formula
- Definition of physical value range (e.g. -20 .. +300 km/h)
- Calculation of precision (e.g. 0.1 km/h)
- Selection of implementation type (e.g. sint16)
- Calculation of implementation value range

ASCET generates safe code for fixed-point arithmetic:
- Protection against overflow & division by zero
ECU Code Generation with ASCET
ASCET Support for Fixed-Point Implementation
ECU Code Generation with ASCET
ASCET Support for Fixed-Point Implementation

```c
/* public process [] */
void MODULE_BDE_EXAMPLE_IMPL_process(void)
{
    /* process: sequence call #1 */
    E = (A + B) * C / D;
}
```
Automatic Generation of Overflow Protection
Automatic protection against division by zero
ASCET can be configured to meet different objectives regarding RAM/ROM & Runtime constraints.
ECU Code Generation with ASCET
ASCET Code Generation Facts

- ASCET’s code generator is certifiable
  - E.g. IEC 61508 SIL 3 “Fit for Purpose” certificate for ASCET V5.1.3

- ASCET can generate up to 100% MISRA-C:2004 compliant code

- Fastest turnaround times
  - Rebuild all for complete ESP application SW (>1 MLOC) in less than 10 min

- Many embedded targets (processor/compiler) supported
  - Adaptation to other targets easily possible

- Arithmetic services
  - Support for mapping arithmetic operations to highly optimized library services
− BMW AG

“Because of modelling guidelines the demand for runtime, ROM and flash is almost identical for automatically generated C-Code and C-Code which is coded by hand.”

Ref: BMW/Department Head Electronic Development, Mr. J. Hauser
− ETAS Competence Exchange Symposium

− Robert Bosch GmbH – GS (Gasoline Systems)

“For complex models, the deployment of ASCET V5.1 may yield time savings in the area of up to 30 percent.”

Ref: BOSCH-GS, Mr. Nicolaou – ETAS RealTimes
− BMW AG

“Because of modelling guidelines the demand for runtime, ROM and flash is almost identical for automatically generated C-Code and C-Code which is coded by hand.”

Ref: BMW/Department Head Electronic Development, Mr. J. Hauser
− ETAS Competence Exchange Symposium

− Robert Bosch GmbH – CC (Chassis Control)

“Better than Simulink because your model is always a discrete model – no fear of transforming that to ECU code”

Ref: Mr. Klug, BOSCH-CC - 2005
ECU Code Generation with ASCET
Customer Testimonials – Development Efficiency

- Volvo Truck
  “We beat the other group using TargetLink when moving to the new Freescale MPC55xx target by a month.”

  Ref: Mr. Elgh, Volvo Truck – 2005

- Ricardo
  “Hourses for courses: If you need to identify the control strategy, use Simulink – if you want ECU code use ASCET”
  “We put a factor 4 on project costs with customers who want us to do an ECU project not with ASCET but with Simulink”

  Ref: Mr. Andrew Smith, Ricardo - 2006
ECU Code Generation with ASCET

Major usage scenarios

- **ASCET as additional programmer**
  - Use case: Delivery of SW components
  - Provides file list for external make environments
  - Provides the set of .c and .h files

- **ASCET as integrated project environment**
  - Creates executable hex file
  - Integrates operating system
  - Contains make files for complete build
  - Extracts addresses for later Measurement & calibration

![Diagram showing the flow of ECU code generation with ASCET.](image)
ECU Code Generation with ASCET
Customer Example: Dual Clutch Transmission Control

– Developed a transmission control unit from scratch in four months
  – ASCET used as SW development integration platform
  – SW development even finished ahead of schedule
– ASCET continuous-time plant model allowed early testing of algorithms before test vehicle was available
– ECU based on MPC555

Dr. Richard Taylor, Technical Director, VOCIS Driveline Controls:
„Ascet tool chain proved to be as reliable and robust as believed, and was a key element in achieving project success.“
ASCET in ECU Development Tool Chains

Outline

− Modeling with ASCET
  − Control Models, Plant Models
  − AUTOSAR Support

− ECU Code Generation
  − From an abstract function model to ECU production code
  − Generate a complete ECU software

− Tool Chain Integration
  − What is needed beyond modeling and code generation
  − How does ASCET integrate in tool chains

− Conclusion
Tool Chain Integration
There is much more than modeling & generating ECU code...

- Requirements Management
- SW Model Exchange
- Open Data Model & Automation API
- Validation & Test Environment
- Version & Configuration Management
- ECU SW Build
- ECU Calibration
- SW Model Exchange
- Reusing software is a huge productivity boost
  - ASCET models are portable
Tool Chain Integration
ASCET models are portable

- ASCET supports reusability by separation of concerns
  - Separate information is stored separately and does not pollute the function
  - Can easily change data or implementation to generate a variant
Tool Chain Integration
ASCET models are portable

ASCET supports reusability by separation of concerns
- Separate information is stored separately and does not pollute the function
- Can easily change data or implementation to generate a variant

Enable/disable/change model behavior based on system constants

```c
#if SysConst == 42 /* Something */
#else
/* Something else */
#endif
```
Tool Chain Integration
Some Paradigms of Professional Software Development

- Reusing software is a huge productivity boost
  - ASCET models are portable
- Software development is an incremental task
  - ASCET helps you to manage different versions of your model
  - ASCET finds and visualizes differences between models
Tool Chain Integration
ASCET helps you to manage different versions of your model

- ASCET can act as a Front-End for Configuration Management Systems
  - Transparent integration via menu, toolbar and overlay icons
  - Full access to version attributes of ASCET elements
  - No more switches between ASCET and Configuration Management System
  - Ideal support for distributed development

Jan Franke, Senior Manager Powertrain SW Development, Audi AG:
„ASCET has become the single do-all application for all daily work routines.“
„[The achieved] automation of routine tasks, results in a high degree of efficiency and safety.“
Tool Chain Integration
ASCET finds and visualizes differences between models

- Find and visualize differences in
  - Graphical (block diagrams, statemachines) and textual models (ESDL, C)
  - Configurable reporting

Peter Döbler, Powertrain Function Development, BOSCH:
„The tool ASCET-DIFF puts an end to this time-intensive task [the manual change review of model updates]. It helps function developers and engineers to spot the changes on a model quickly and efficiently.“
Tool Chain Integration
Some Paradigms of Professional Software Development

- Reusing software is a huge productivity boost
  - ASCET models are portable
- Software development is an incremental task
  - ASCET helps you to manage different versions of your model
  - ASCET finds and visualizes differences between models
- Testing software is as important as generating it
  - ASCET allows early SW tests on the PC
  - ASCET will support you in evaluating the quality of your tests
**Tool Chain Integration**

ASCET allows early SW tests on the PC

- Run/test the same model on different targets
- Use “compare computing” to qualify that PC simulation results are equal to Production ECU results
  - High productivity gain due to faster setup times for PC Simulation
  - High cost savings as tests can be performed at the developer’s desk instead of in the lab or on the test track
- This approach is used at BOSCH Chassis Control
Tool Chain Integration
ASCET will support you in evaluating the quality of your tests

- How do you determine when you have tested enough?

- ASCET-MCA (Model Coverage & Analysis)
  - determines which parts of your model have been covered by your tests
  - identifies blank spots of your tests
  - visualizes its results by coloring the model

- Further use case
  - Automatic introduction of debug variables

- ASCET-MCA is planned to be available with ASCET V6.2 (Q3/11)
Tool Chain Integration
Some Paradigms of Professional Software Development

- Reusing software is a huge productivity boost
  - ASCET models are portable
- Software development is an incremental task
  - ASCET helps you to manage different versions of your model
  - ASCET finds and visualizes differences between models
- Testing software is as important as generating it
  - ASCET allows early SW tests on the PC
  - ASCET will support you in evaluating the quality of your tests
- Software development spans often several companies
  - ASCET allows you to collaborate with others
Tool Chain Integration
ASCET allows you to collaborate with others

- MATLAB/Simulink models can be imported in ASCET
  - Basic transformation rule set for most commonly used blocks available
  - Customizable mapping of referenced library blocks possible (e.g. MBFS)

Stefan Trinkert, Powertrain Function Development, BOSCH:
„The transformation of a complex diesel project took only a few hours, whereas the manual migration of comparable projects used to require a number of days. Staff members who had no prior experience with ASCET were intuitively able to handle the tool immediately.“
Tool Chain Integration
ASCET allows you to collaborate with others

- Integrate a compiled ASCET model (DLL) in other simulation environments

- ASCET model (parts) as Simulink S-function

- Use Case at BOSCH Chassis Control:
  - Provide ESP control algorithm to OEM for early validation (before ECU is available)
  - Intellectual Property of ESP control is protected (compiled model DLL)
Tool Chain Integration
Some Paradigms of Professional Software Development

– Reusing software is a huge productivity boost
  – ASCET models are portable
– Software development is an incremental task
  – ASCET helps you to manage different versions of your model
  – ASCET finds and visualizes differences between models
– Testing software is as important as generating it
  – ASCET allows early SW tests on the PC
  – ASCET will support you in evaluating the quality of your tests
– Software development spans often several companies
  – ASCET allows you to collaborate with others
– Development tools will have to integrate with other tools
  – ASCET provides all the open interfaces you need
  – ETAS supports you with customer-specific solutions
Tool Chain Integration

ASCET provides all the open interfaces you need

- ASCET’s entire model database is stored as XML files in the file system
  - Really open format – documented schema is provided with tool

- ASCET imports/exports open formats including ASAM, AUTOSAR and XML

- Remote Control by Automation-API (based on WebServices)
  - Database control incl. import/export
  - Code generation, compile, build/rebuild
  - Experiment control (e.g. start/stop, log)
Tool Chain Integration
ASCET provides all the open interfaces our partners need

- Requirements Tracing (OMNI Team DOORS-ASCET-Link)
  - Linking ASCET elements with DOORS requirements
  - Requirements coverage checks & change impact analysis

- Modeling Guideline Checker (Match Technologies Assessment Studio)
  - Definition and checking of ASCET modeling guidelines
  - E.g. catalog for usage-dependent MISRA rules

- Static Code Analysis Integration (AbsInt a³ (aiT & StackAnalyzer))
  - Safe upper bound on task/process execution times and stack usage
  - ASCET generates annotations and controls analysis

- Schedulability Analysis (Symtavision SymTA/S)
  - System-level timing analysis & optimization
  - ASCET generates SymTA/S project
ASCET in ECU Development Tool Chains
ETAS supports you with customer-specific solutions

- BOSCH-DGS uses MSR-SW based build tool chain for MEDC17
  - ASCET-generated C-Code modules need to be described in MSR-SW

- Read, Edit, and Write MSR-SW information in ASCET
  - Structure, Parameters, Variables
  - Implementations
  - Formulas

- MEDC17 Add-on Package available for all ASCET customers
  - MSR-SW data generation & management
  - Optimized service library
  - ECU-specific code generation adaptations
ASCET in ECU Development Tool Chains

Conclusion

- ASCET has a proven track record in ECU SW development
  - In small projects as well as ...
  - ... in large-scale developments with over 1000 SW developers

- At OEMs as well as ...
  - ... at suppliers and engineering service providers.

- ASCET can also be the backbone of your ECU SW development

- ETAS is a cooperative and reliable partner for your ECU SW development projects
Vielen Dank

Thank you

Merci

Vielen Dank