Model-based Electric/Electronic Development
from Architecture Design to Series-Production Readiness
Model-based Electric/Electronic Development from Architecture Design to Series-Production Readiness

PREEvision offers you an integrated approach for the entire electric/electronic development process in the automotive industry. Moreover, the proven Systems Engineering principles of abstraction, modularization and reuse are supported.

- **Product line approach:**
The product line approach forms the basis for defining product variants and reusing system components.

- **Automotive standards:**
Automotive standards (such as AUTOSAR, FIBEX, DBC, LDF, KBL, RIF) are supported, which enable data exchange and integration in existing development environments.

- **System optimization:**
Calculation functions (metrics) can be defined by the user. The metrics lead to optimized architectures, which in turn lead to lower product costs.

- **Functional safety:**
Functional safety according to ISO 26262 for electronic systems is supported by integrated design and analysis functions. This leads to reduced development effort and continuous traceability in safety-related systems.

- **Increased efficiency in development:**
Efficiency is increased by comprehensive functions for collaboration in teams.

- **Planning and tracking in development:**
The integrated product, release and change management functions support planning and tracking in development.

- **Risk reduction in development:**
Automatic validation and consistency checks reduce risk even in early development phases.

- **Transparency in development:**
All development artifacts are available to all participants in their most current states. Coordination processes are tool-supported.

Use case areas:
PREEvision supports you in:

- Research and concept development, design of the E/E architecture and its evaluation, optimization and documentation.

- Series-production development, requirements specifications, integrated creation of test specifications, component and network specification and in defining and maintaining logical and physical interfaces.

Advantages:

- **Intuitive, graphical user interface:**
The intuitive user interface supports advanced diagrams for modeling the various layers of electronic systems (e.g. network, hardware and software layers).

- **Practice-proven integrated data model:**
The continuous data model was developed in cooperation with leading automotive OEMs and suppliers.
PREEvision user interface

1. Model view for navigation in the model
2. Outline for displaying the relevant subset of the graphical model
3. Geometry diagram
4. Hardware network diagram
5. Property view for editing all attributes
Development of electronic systems:

PREEvision supports modeling of E/E systems by abstraction and modularization. The following abstraction layers are supported:

- Product goals (customer features and requirements)
- Logical architecture
- Software architecture (compliant to AUTOSAR)
- Software implementation (including MATLAB®/Simulink®-integration)
- Communication
- Hardware architecture (network and component view)
- Electric circuit and wiring harness
- Hardware geometry

The artifacts of the different abstraction layers are interrelated via mapping objects.

Integrated systems engineering:

PREEvision supports integrated systems engineering by:

- Integrated requirements management
- Product line and variant management
- Automatic consistency checks to assure quality of the designs
- Complex metrics (calculation functions) for evaluating alternative architecture concepts
- Configurable report generator
- Analysis tools for safety-related systems
- Version and life cycle management
- Collaboration platform for distributed development projects
- Product and release management
- Change management
- Test data management
- Roles and rights concept
<table>
<thead>
<tr>
<th>Product Line and Variant Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency Checks and Metrics</td>
</tr>
<tr>
<td>Report Generator</td>
</tr>
<tr>
<td>PREEvision Collaboration Platform*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Architecture</td>
</tr>
<tr>
<td>SW Architecture and Implementation</td>
</tr>
<tr>
<td>Communication</td>
</tr>
<tr>
<td>Hardware Network and Component Architecture</td>
</tr>
<tr>
<td>Electric Circuit and Wiring Harness</td>
</tr>
<tr>
<td>Hardware Geometry</td>
</tr>
</tbody>
</table>

PREEvision *optional

PREEvision overview
Graphical diagrams support modeling of the different architecture layers of a product line. Products can be formed as subsets of a product line with the help of the integrated variant management.

The reuse of engineering artifacts is also possible across product lines. Data exchange with other tools is supported by import and export filters for many industry standards.
The hardware geometry layer shows all installation spaces and mounting locations in the vehicle in 2.5D representation. Wire routing paths are modeled as line segments with disconnection and connection points. The vehicle's geometric model contains real geometric dimensions with lengths and cross sections. All elements of the geometry also have 3D coordinates. Coordinated adjustment of the geometry in relation to external CAD tools or data models is also technically feasible.

The wiring harness layer contains all line connections, plug connectors, splices, disconnection points, pins and assembly materials of the hardware architecture. It is possible to model copper wire, aluminum wire and optical conductors. The wiring harness is visualized in a wiring harness diagram.

The electrical-logical connections between all physical components are modeled on the electric circuit layer. For example, a CAN bus that is visualized by a line in the network diagram is depicted by its physical implementations in the electric circuit diagram. This layer is also used to model the power supply architecture. All hardware components are supplied with electrical power or are connected to vehicle ground via different terminal types. Special power supply components may be modeled directly (e.g. with fuses and relays).

This layer specifies all physical signals in the hardware network. CAN, LIN, and FlexRay bus systems are supported. That is, signals, PDUs, frames and schedules may be specified. The integrated signal router makes it possible to automatically generate communication artifacts.

This layer describes all ECUs, sensors and actuators and their networking via bus systems. Supported here are ring, star and line topologies. Communication over conventional connections is also specified. Also modeled in the network diagrams are connections for power supply and grounding concepts. The network diagram abstracts from the wiring diagram (multiple wire connections) are visualized by one line.

Customer features describe the planned range of a product line from the customer perspective, e.g. the features of a vehicle. In addition, all types of functional and nonfunctional requirements are also formulated, hierarchically arranged and interlinked on this layer.

The logical architecture defines an abstract graphical functional network of the later implementation in software or hardware. The logical architecture encompasses the specifications of logical components (such as sensor, actuator and logical functions). In addition, interfaces and connections are specified. Elements of the logical architecture can be mapped to the higher customer feature layer and to the underlying software or hardware architecture.

The software architecture encompasses the specifications of software components, their ports and port interfaces, and their interconnections. The software architecture supports the AUTOSAR methodology. The software components can be mapped to the hardware network architecture.

The implementation layer manages all implementation artifacts of the software components. In particular, this layer supports model-based behavioral modeling with MATLAB®/Simulink®. Furthermore, storing models and libraries (mdl-files, m-scripts etc.), the Matlab®/Simulink® integration also ensures consistency between the software architecture and the behavioral model concerning ports and connections.

The hardware component architecture describes all internal aspects of an ECU with an electric circuit diagram, e.g. power supply, ground connections, microcontrollers, bus interfaces, plug connectors, memories, housings and printed circuit boards.

The logical architecture defines an abstract graphical functional network of the later implementation in software or hardware. The logical architecture encompasses the specifications of logical components (such as sensor, actuator and logical functions). In addition, interfaces and connections are specified. Elements of the logical architecture can be mapped to the higher customer feature layer and to the underlying software or hardware architecture.

The system software architecture encompasses the specifications of software components, their ports and port interfaces, and their interconnections. The software architecture supports the AUTOSAR methodology. The software components can be mapped to the hardware network architecture.

The implementation layer manages all implementation artifacts of the software components. In particular, this layer supports model-based behavioral modeling with MATLAB®/Simulink®. Furthermore, storing models and libraries (mdl-files, m-scripts etc.), the Matlab®/Simulink® integration also ensures consistency between the software architecture and the behavioral model concerning ports and connections.

The hardware component architecture describes all internal aspects of an ECU with an electric circuit diagram, e.g. power supply, ground connections, microcontrollers, bus interfaces, plug connectors, memories, housings and printed circuit boards.
The description of all significant aspects of an E/E architecture with the help of the graphical or table-supported editors leads to a “150%” model. This represents the maximum content of a vehicle concept and contains all installable E/E equipment features of a vehicle product line. For example; A model might include different and mutually exclusive drive concepts. Alternative networking concepts, electrical system configurations and topologies can also be modeled.

The architect can define the success criteria for an E/E architecture. For example, the ‘adjustment dials’ for the success of an architecture might be based on different optimization goals such as weight, cost, bus load, etc. A multi-dimensional decision network can be set up to evaluate the architectural concept. This makes it possible to evaluate one architecture alternative in comparison to other alternatives.

PREEvision is used for model-based conceptualization, development and evaluation of E/E architectures. The goal is quick conceptualization and evaluation of architecture alternatives over all relevant technical layers.

The description of all significant aspects of an E/E architecture with the help of the graphical or table-supported editors leads to a “150%” model. This represents the maximum content of a vehicle concept and contains all installable E/E equipment features of a vehicle product line. For example; A model might include different and mutually exclusive drive concepts. Alternative networking concepts, electrical system configurations and topologies can also be modeled.

The architect can define the success criteria for an E/E architecture. For example, the ‘adjustment dials’ for the success of an architecture might be based on different optimization goals such as weight, cost, bus load, etc. A multi-dimensional decision network can be set up to evaluate the architectural concept. This makes it possible to evaluate one architecture alternative in comparison to other alternatives.

PREEvision is used for model-based conceptualization, development and evaluation of E/E architectures. The goal is quick conceptualization and evaluation of architecture alternatives over all relevant technical layers.

The description of all significant aspects of an E/E architecture with the help of the graphical or table-supported editors leads to a “150%” model. This represents the maximum content of a vehicle concept and contains all installable E/E equipment features of a vehicle product line. For example; A model might include different and mutually exclusive drive concepts. Alternative networking concepts, electrical system configurations and topologies can also be modeled.

The architect can define the success criteria for an E/E architecture. For example, the ‘adjustment dials’ for the success of an architecture might be based on different optimization goals such as weight, cost, bus load, etc. A multi-dimensional decision network can be set up to evaluate the architectural concept. This makes it possible to evaluate one architecture alternative in comparison to other alternatives.
PREEvision enables this benchmarking of alternative architectural concepts based on customer-specific metrics for all modeling layers. The results of this performance comparison can serve the architect as a decision-making tool for choosing an architecture concept. Moreover, it provides important information that is useful in further refinement of the architecture over the course of the development process.

Based on the “150%” model, the variant management tools allow architects to structure the different architecture alternatives in the context of a model series or platform development. The goal is to develop an E/E system that is as cost-optimal and robust as possible.
PREEvision is used for requirements engineering of E/E systems and their components. The goal is to define customer features and requirements and automatically generate specification documents.

With PREEvision, customer features and requirements can be created, organized and managed in versions easily and quickly. Import filters are available for Excel and RIF. Customer features and requirements are described with formatted text, graphics or tables. This is done using the powerful integrated functionality of Open Office. User-specific attributes can also be defined.
Requirements can be mapped to one another or to any desired artifacts of the product line model, e.g., to the logical architecture or hardware architecture. Relationships can also be established to test, safety analysis and planning artifacts. This permits continuous traceability of requirements over the entire development process right up to the testing phase.

The integrated report generator supports the function or component developer with automatic creation of documents, e.g., of system specifications or specifications for software and hardware components. The graphical diagrams of all modeling layers and the attributes of all model artifacts can be automatically integrated into the generated documents.
PREEvision supports item definition, hazard and risk analyses, system safety concept development, system and component design, safety analyses as well as verification and validation. The safety engineer derives safety goals from a hazard and risk analysis of the system. The safety goals are then refined, and system functions and components are assigned. The functional and technical safety strategies are also defined in this way.

The integrated tool functions for creating an FMEA (Failure mode and effects analysis) ensure that critical failure types of components are evaluated and that the necessary improvement actions are defined and implemented in a traceable way in the architecture, requirements and test cases.

PREEvision enables an ISO 26262 compliant approach to modeling and analyzing functional safety concepts while designing the E/E architecture.
The AUTOSAR system architect specifies types of application software components and compositions in the Software Architecture Library. Graphical diagrams support the definition of software architectures and the connection of ports. Numerous engineering features that simplify work for the AUTOSAR developer are available, such as automatic port connectors or refactorings of existing components.

The hardware network architecture can be graphically modeled as well. Mapping of the software components to the hardware components is supported by graphical and table-based views. Interrelationships across hierarchies in the software architecture can be represented in software system diagrams. PREEvision supports different AUTOSAR versions for import and export.

PREEvision supports specification of an AUTOSAR-compliant system architecture. Available for this purpose are graphical diagrams of the software architecture, the ECU network and numerous engineering features.

Mapping software components to the hardware
PREEvision consists of the following products:

- PREEvision Architect
- PREEvision Function Designer
- PREEvision Electric Designer

and product options:

- vTESTcenter
- PREEvision Collaboration Platform

These tools let you set up the optimal configuration for your use cases.

**Example:**
PREEvision Function Designer and the option PREEvision Collaboration Platform are used to create a function and component development environment for a team in series-product development. The Collaboration Platform includes product and release management.

Ask us about PREEvision! We would be glad to advise you on how to set up a tool configuration for your specific use cases.

▶ E-Mail: sales@vector.com
**PREEvision Architect** supports all modeling layers for product lines.

**PREEvision Function Designer** and **PREEvision Electric Designer** support sub-areas.

**vTESTcenter** for test data management and **PREEvision Collaboration Platform** for team collaboration and product and release management are optional extensions.

The roles and rights concept lets you set up different views and authorization levels for accessing individual artifacts, e.g. model layers, for different user groups.
More Information

> News
> Products
> Demo Software
> Support
> Workshops
> Addresses

www.vector.com