Electric-electronic architectures (E/E) in vehicles – What’s the first thing that comes to mind? Complex systems which have been becoming increasingly more extensive for years? Where an ever-greater number of components build upon one another and are dependent on one another? Where the individual components are subject to continual adjustment and expansion during development? And that it’s becoming increasingly difficult to maintain an overview for recognizing correlations and dependencies? If so, you should take a closer look at the advantages of an integrated development tool and requirement management.

Whether it’s networked systems, convenience electronics and multimedia, driver assistance systems, autonomous driving, connectivity or cloud applications, the number of communication systems in vehicles is increasing relentlessly and E/E architectures are becoming more complex and diversified. Despite this, dependencies need to be quickly and easily identifiable when developing E/E systems – everyone involved has to be certain at all times as to where the modification, adjustment or expansion of a system requirement needs to be taken into account. Questions like: “Which components are affected by modified requirements?” and “What test cases need to be adjusted or run again?” must be answered without delay. If safety requirements or legal regulations change, the places in the system where they must be adjusted have to be traceable. Classic requirement management just doesn’t cut it here anymore.

The Limits of Classical Requirement Management

With classic requirement management, requirements were originally collected in text editing or spreadsheet programs without any help from special tools. Traceability was as good as impossible using this method. It wasn’t until independent, self-contained tools for requirement management came along that it was possible to relate requirements in a structured way and with one another – something which ensures general traceability. By linking to external data or other content, the limits of tools can also be surpassed. The problem here is that these links often quickly become obsolete or invalid. This is especially the case when working across multiple tools or databases, where links cannot be defined on the atomic level and are difficult to maintain.

While a small team used to be sufficient for creating requirements, the complexity of today’s systems necessitates the management of an interdisciplinary team – complete system or component specifications don’t come about until system and function developers, architects responsible for components, communication designers, test managers and other areas of responsibility work together. Classic requirement management systems quickly reach their limits here as well.
Integration as the Key to System Mastery

Most vehicle functions provided by E/E systems concern control, regulation, monitoring and diagnostic functions. They interact with the mechanical vehicle components using sensors and actuators to be installed in various geometric locations in the vehicle. This means that E/E systems are also geometrically distributed systems which work closely together in a distributed function network, such as driver assistance functions with the functions of the drive, steering and brake systems.

This is why comprehensive E/E architectures with complex and intermeshed function and control unit networks can only be efficiently mastered in a computer-aided development environment with a central database. Without a central E/E database like this, the expenditure for internal alignment and maintenance of redundant and possibly inconsistent data and requirements quickly rises.

This results in high development costs, more time being required and, in the worst case scenario, errors which are first noticed out in the field. The only way this enormous task can be overcome is for an organization to maintain an overview of all development artifacts and their correlations through common storage of the artifacts, visualization, computer-aided evaluations and consistent traceability. With the combination of standalone authoring tools for the E/E system and requirement management tools which are independent of it still being widely used today, however, it can only be implemented with restrictions or not at all. This is because cross-tool and cross-division cooperation, as well as relation on the atomic level, are not possible with it. Point-to-point connections often arise here. Integration of the individual tools is partially implemented using standards such as OSLC (Open Services for Life cycle Collaboration); the integrated artifacts are then visible in the different tools and can also be edited to a degree there, but the relationships beyond database boundaries which have arisen in this way cannot be reliably versioned and archived [Figure 1].

Redundant data management with substitution objects using cyclical import/export and update processes has the disadvantage of an inconsistent data model. This results in noticeable tool limits, a variety of different architectures (such as Rich Client and Web Client) and a variety of different concepts for release, versioning, backup and migration in the tools. A large integration and operation expenditure arises here, yet the solution only partially satisfies the actual requirements.

Figure 1: A disadvantage of heterogeneous development environments without consistent data models: Noticeable limits to stand-alone authoring tools for E/E systems and requirement management.
Integrating Requirements directly into the Development Process

The solution to this dilemma is provided by the next evolutionary stage in requirement engineering: integrated requirement management within the E/E development environment as implemented in the PREEvision E/E architecture tool. Here, requirement management is integrated directly into the development process [Figure 2].

Involved persons from all areas work together on a database, which provides a seamless and transparent view of the overall system and enables structured cooperation. Thanks to the model-based approach of PREEvision, the various different E/E artifacts and their relationships are dissolved and managed in great detail. For a variety of different abstraction levels and roles, consistent and valid documents are created in this environment. During the development phase, changes can be responded to efficiently and quickly at any time as well. Fundamental requirement management functions are available for this purpose:

- Collecting requirements
- Hierarchical structuring
- Linking and tracking
- Validating
- Importing and exporting (ReqIF, RIF and Excel)
- Creating system specifications or performance specifications

In addition, integrated requirement management enables seamless interaction between the requirements and all development artifacts in the model. Not only the requirements, but the development artifacts themselves as well, serve as requirements in the integrated approach. These are linked with requirements or embedded directly in requirements using placeholders.

The integration of individual property values of an artifact is also possible. In E/E development, the concrete artifacts are often defined in much more detail than is common with conventional requirements. For example, the properties of the individual pins of a plug are seldom described in classic requirement management. With the integrative approach, however, they can be collected automatically and part of the requirement specification can be used. The placeholder principle, which works in a similar way to field functions in text editing programs, ensures that the current values are always accessed automatically in the dynamic development process here.

![Figure 2: Modeling levels of the PREEvision E/E development tool.](image-url)
In addition, a rights management system which provides monitored access and role-dependent editing rights is required. Ultimately, highly detailed version management is required in order to ensure unique and reproducible development statuses. Joint simultaneous work on the development and management of E/E systems in vehicle and automobile manufacturing is supported in PREEvision with a collaboration platform in a three-tier architecture. Its functions are designed for structured creation and implementation of complex E/E projects in large organizations. The central database ensures a homogeneous dataset, which enables organization-wide multi-user operation. Project and development teams have shared access to the necessary data and work simultaneously without data conflicts. Expensive processes for merging datasets are thus eliminated. Versioning is carried out with functions for checking in and checking out. As a result, requirement development also remains traceable over time [Figure 4].

Developing Requirements Seamlessly, Consistently and Completely
All software and hardware components in the development model can be seen as requirements and used in system specifications or performance specifications. With integrated requirement management, you continually receive a current and detailed requirement specification, rather than a text-only description. Integrating and augmenting details from the E/E development data ensures consistency, completeness and comprehension.

The model-based tool environment with integrated requirement management permits top-down, bottom-up and middle-out approaches here, where the requirement specification is either classically based on requirements and carried over to concrete E/E artifacts or derived directly from the E/E artifacts. Mixed operation with requirements and specifications defined in concrete E/E artifacts is also possible, offering advantages from both worlds. For secure data exchange with suppliers, both requirements and any other desired E/E model artifacts are exported in the ReqIF exchange format over the standardized ReqIF interface of PREEvision. They are then available to other tools for integration [Figure 3].

Developing Requirements Together and Simultaneously
Many users with different roles don’t just work on a single component or a single system on a single team. This is why it must be possible to access, create and manage E/E artifacts, requirements and test cases across the organization. In addition, a rights management system which provides monitored access and role-dependent editing rights is required. Ultimately, highly detailed version management is required in order to ensure unique and reproducible development statuses. Joint simultaneous work on the development and management of E/E systems in vehicle and automobile manufacturing is supported in PREEvision with a collaboration platform in a three-tier architecture.

Its functions are designed for structured creation and implementation of complex E/E projects in large organizations. The central database ensures a homogeneous dataset, which enables organization-wide multi-user operation. Project and development teams have shared access to the necessary data and work simultaneously without data conflicts. Expensive processes for merging datasets are thus eliminated. Versioning is carried out with functions for checking in and checking out. As a result, requirement development also remains traceable over time [Figure 4].

Developing Requirements in a Transparent and Process-Reliable Way
Thanks to the model-based, integrated approach of PREEvision, persons responsible for components can also view the requirements of the system level and access a concrete implementation. As a result, the requirements which are implemented by a specific component remain identifiable. Conversely, requirements which have not yet been covered can be identified in this way. The integration of test data management also identifies the test cases.
Requirement engineers, for example, can no longer work in a specialized tool which is purely custom tailored to their needs. These aspects were identified in PREEvision and solved using suitable mechanisms: Perspectives enable a special view of the system data for dedicated user groups and provide a user-specific interface (use case-driven GUI) which supports the respective work flow. Scopes reduce the visible data extract and only focus on the required artifacts – complexity is greatly reduced here [Figure 5].

Visualizations ensure easy access to the system, facilitate comprehension and represent correlations in a transparent way. Examples here include technical diagrams for a wiring harness or network and behavioral diagrams like use case diagrams. The reuse of requirements, functions and other E/E artifacts accelerates development noticeably and ensures data consistency, so the recreation or copying of similar or identical artifacts is avoided. Last, but not least, validators and testing mechanisms ensure consistency and offer automated solutions for identified problem areas – as soon as data is entered.

Eliminating Disadvantages of the Integrative Approach

The increased complexity of the development process through the embedding of all data relevant to the E/E system and the high number of users and roles involved may be considered disadvantages of the integrative approach. User acceptance is also a challenge.
A consistent tool in which users can access all the data relevant to them provides clarity and comprehension for all roles involved with system development. It makes the complexity of the overall E/E development process controllable and enables conflict-free and consistent cooperation – quickly, flexibly and efficiently. Current, detailed and always traceable requirements are an important key to the success of an E/E system, as their seamless integration prevents tool breakdown and enables the traceability of all data in an E/E system. PREEvision has been implementing this approach for years. It has proven itself in productive use and supports all E/E development in a consistent, model-based and future-proof way.

**Summary and Outlook**

**Figure 5:** With the integrative approach of PREEvision, mechanisms like perspectives and scopes reduce complexity and increase both efficiency and user acceptance.

**Author**

Robert Rotter  
Vector Informatik GmbH  
Senior Product Management Engineer PREEvision

Originally published in “Elektronik automotive” magazine issue November 2017