Autosar for Agricultural Electronics

The agricultural machinery market demands ever more powerful customer functionalities like those of Precision Farming. The high diversity of products with low production volumes results in a high proportion of software development costs in systems. In the automotive industry, which faces a similar situation regarding the costs, the Autosar standard facilitates the reduction of development time and expenses. To date, Autosar has not been ready for the agricultural machinery market because the network standard Isobus was not supported. To bridge this gap, ITK Engineering has developed a solution which expands Autosar with Isobus.
functionalities into systems, using an
optimised distribution of work between
manufacturers, suppliers and tool
vendors.
Software and electronics in agricul-
tural machinery present a compelling
purchase argument as they directly
influence productivity: In fact, agricul-
tural machines must enable efficient
accomplishment of farming tasks and
software can support this. The market
features a large diversity of application
areas, depending on, for example, par-
ticularities in different countries. Devel-
opers therefore require profound domain-
specific knowledge. Not only are differ-
ent types of vehicles such as tractors or
specific harvesters required, but imple-
ments further cover farming tasks like
fertilising, seeding or spraying. Agricul-
tural machinery manufacturers thus
address a large market diversity of pro-
ducts with intensive specialisations.

Innovations in agricultural machinery
are driven by technological advances in
sensors, automation and interconnectiv-
ity [4]. Today, sensors go far beyond
the measurement of simple values and
gather complex environment variables
like the content of nutrients in the soil.
Not only is guidance supported with pre-
cise positioning for vehicle and imple-
ment, but even automated steering is
provided. In this way, information gath-
ered in the vehicle allows for functionali-
ties with manifold settings for farming.
Moreover, interconnectivity enlarges
functional ranges through sharing of
information beyond single vehicles, so
that farmers can manage agricultural
tasks. This is the basis for Precision
Farming, where field information can be
gathered from intelligent equipment and
be analysed and evaluated centrally so
that planning of activities on the field is
improved [5]. Such functionalities how-
ever demand higher amounts of software
in systems. Agricultural machinery
thereby sells in low volumes. Thus, the
proportion of software development
costs in agricultural machinery systems
is even higher than in the automotive
industry [6]. Software development and
especially software architecture are key
factors for cost savings.

Proprietary operating systems, drivers,
and lack of separation of software mod-
ules may negatively affect software de-
velopment. Proprietary software is pri-
marily conceived for usage in specific
projects, whereas the long-term aspects of
development are secondary. The reuse of
software for electronic drivers is difficult,
as it normally concerns specific hardware
without the abstraction of software from
the latter. Moreover, lack of separation
between customer functionality and soft-
ware for electronic drivers is problematic.
In this way, new customer functionality
requires extensive efforts for software
development of electronic drivers and the
reuse of software is complicated.

Autosar provides a mature concept for
cost savings in software development. In
contrast to proprietary solutions, Autosar
is based on a standardised software
architecture with two main parts: On the
one hand, the application software for
the development of customer functional-
ity, and on the other hand, the Basic
Software which provides standard con-
tenits required by the Application Soft-
ware. The corresponding development
approach enables the efficient integration
of customer functionality into electron-
ics. As a standard that is constantly
refined, Autosar allows for the anticipa-
tion of future software-intensive develop-
ment, so that the mastery of trends like
Precision Farming is sustained [2].

**DEVELOPMENT APPROACH WITH AUTOSAR**

Autosar was founded in 2003 for the
standardisation of automotive embed-
ded software [2]. The development pro-
cess of automotive electronics was origi-
nally characterised by a relationship
between manufacturers, responsible
for the overall system, and suppliers,
accountable for Electronic Control Units
(ECU). This constituted an ECU-oriented
development. Electronic parts of systems
typically were based on proprietary soft-
ware with operating systems conforming
to the OSEK (open systems and their
corresponding interfaces for automo-
tive electronic) standard and for com-
munication in networks conforming
to the CAN (Controller Area Network)
standard. In this way, a multitude of
different platforms evolved, complexity
increased and costs and quality prob-
lems arose.

The fundamental idea behind Autosar
is the quest to find a common denomina-
tor for the usage of software. True to the
motto “cooperate on standards, compete
on implementation”, Autosar makes use
of proven software parts so that software containing customer functionality can be carried on top [3]. The standard brings a change in paradigms, from an ECU-oriented development to a function-oriented development [7]. Development with the standard is divided into two perspectives. FIGURE 1 depicts the development approach of Autosar with SW-Cs (Software Components) containing customer functionality as starting point. The VFB (Virtual Functional Bus) is used as a first perspective for consideration of communication between SW-Cs without electronics. In a second perspective, the deployment of SW-Cs on control units is performed as so-called mapping for consideration of the interactions between functions and electronics. That is where the configuration of the BSW (Basic Software) of each control unit is approached. This development approach is called Autosar Methodology.

The standard primarily focuses on software. The software architecture with the BSW and the consideration of functionality on the VFB anyhow gives Autosar system relevance, as it influences how functionality is delivered in systems. The content is based on three coupled pillars:

- Architecture: A layered software architecture, FIGURE 2, abstracts from control units by means of three layers in the BSW, which integrates established standards for operating systems and networks. The RTE (Runtime Environment) moreover decouples the Application Layer from the BSW.
- Interfaces: The Autosar Interfaces are the key to separation and collaborative development of Application Software and Basic Software. They serve the specification of communication data of the RTE.
- Methodology: Activities and products that are supported by specialised tool-chains, enable the repartition of software development and allow for interoperability in distributed development via the format ARXML (Autosar XML). The methodology is divided into two main perspectives as explained above.

The interplay of these three pillars underlines how the concept affects software development:

- Firstly, the Basic Software can be developed separately from the Application Software. Abstraction of Basic Software from control units improves the independence of software from control units. Basic Software can be configured via specific toolchains and be delivered by Basic Software vendors so that adaptations of software to control units are eased. This especially is an advantage when hardware changes occur. Moreover, Basic Software can be reused across different products and even across manu-

FIGURE 1 Autosar Methodology: function-oriented development with VFB and mapping of SW-Cs on control units with Basic Software [2] (© Autosar)
facturers. Potential reuse of Basic Software parts in turn justifies higher efforts for its development and allows for quality improvements.

– Secondly, development of Application Software can be focused in a specific perspective of the Methodology. Autosar Interfaces with a formal description serve as contract between development parties for Application Software on the one hand and Basic Software on the other. SW-Cs allow for better integration of customer functionality into control units. This makes development of customer functionality more efficient. Further, SW-Cs can be reused on different platforms due to platform-independent implementation and Autosar Interfaces. Autosar standardises software with a focus on customer functionality in its function-oriented paradigm. The approach gives customer functionality a tangible form in terms of the SW-C. The implementation method of customer functionality in turn is not prescribed by the standard. However, the synergy with model-based design offers further advantages for efficiency in development. Early tests of software functionalities allow for a reduction of later test efforts of electronic platforms. Particularly in agricultural machinery, where low production volumes are faced, complications at test benches for electronics can cause substantial cost ratios [9]. To counter this, the consideration of SW-Cs and their communication at VFB allows for virtual integration of customer functionalities in early development phases [10].

From an economical point of view, the migration towards Autosar demands a particular commitment. The standardisation of software architecture and development with the methodology is linked with efforts regarding the adoption of processes and tools. Figure 3 (blue curve). Only after having mastered such transition for development projects, the efficiency can be improved. Compared to proprietary software, the standard offers chances to cope with increasing complexity in electronics and software through the function-oriented development paradigm and separate development of Basic Software. This can avoid the loss of control for development efforts, Figure 3 (red curve).

As Autosar is intended for the automotive industry, its software architecture originally addresses automotive network types like CAN, LIN (Local Interconnect Network) or Flexray. Agricultural machinery for its part is in this context characterised by networks for communication between vehicle and implement as standardised in Isobus [1]. To enable usage of Autosar in agricultural machinery, Autosar thus needs to be expanded with Isobus. This has been accomplished with the following solution.

EXPANDING WITH ISOBUS

Implements have always been central for agriculture. They are commanded by farmers via a terminal and allow to address the diversity of tasks in agricultural machinery. Isobus standardises networks between vehicles and implements and thus represents a milestone for interoperability in agricultural machinery [1]. The need to connect to the vehicle network in agricultural machinery causes a major technical difference on electronic networks compared to...
automotive industry: Dynamic communication with changing communication nodes over time must be enabled for networks in agricultural machinery, whereas the automotive industry classically uses static networks. **FIGURE 4** depicts a standard network in agricultural machinery based on the example of a tractor with implement. For the operator of agricultural machinery, Isobus is mainly characterised by the Virtual Terminal for the interaction with implements. From a technical perspective, Isobus is based on SAE J1939 (Society of Automotive Engineers), the network standard in utility vehicles, which in turn is built on the automotive standard CAN [1].

Autosar, as a native automotive standard, did not initially include standards for dynamic network communication. The VFB of the standard stands symbolically for the static configuration of networks at design time. Only since 2009 have concrete actions been taken for the usage of Autosar in utility vehicles with SAE J1939 [11]. With versions 4.0 and 4.1, the communication stack of Autosar has been extended by special modules for SAE J1939, such as the J1939 TP (Transport Protocol), J1939 RM (Request Manager) or J1939 NM (Network Manager). With this, a foundation for the usage of Autosar in utility vehicles is given.

Autosar so far was not usable for electronics in agricultural machinery. The reason is that Autosar did not include network communication with Isobus, as presented previously. Autosar CDDs (Complex Device Drivers), however, offer a generic basis for expansion of the standard. They enable the integration of software that is not part of the standard within specific modules. These modules are coupled with the Autosar software via particular interfaces.

ITK Engineering has developed a solution to support Autosar with Isobus in cooperation with leading agricultural machinery partners. The Autosar architecture therefore is expanded on the basis of SAE J1939. As shown by the blue boxes in **FIGURE 5**, the main parts of Isobus are realised as Autosar CDDs. This expansion of Autosar docks on the communication services and the modules of SAE J1939 via Standardized Interfaces (another interface category for software) of Autosar. In this way, a standard software architecture is delivered which covers the usage of implements in agricultural machinery, from monitoring to control of different tasks. The Isobus Application Services addressed in the solution are [1]:

---

**FIGURE 4** Typical network topology in agricultural machinery based on the example of a tractor with implement [1] (© S. Oleg | V. Dmitry | Shutterstock | Isobus | ITK)

**FIGURE 5** Software architecture for expansion of Autosar version 4.1 (or higher) with Isobus (ISO 11783) (© Autosar | ITK)
Virtual Terminal Client (ISO 11783-6): functions that provide capabilities for monitoring and commanding the implement at the terminal.

Task Controller Client (ISO 11783-10): functions that provide capabilities for managing farming tasks (machinery, operator, products, etc.), documenting and controlling them.

File Server Client (ISO 11783-13): functions that provide storage for data and means to use the data.

Beyond the advantage of improved interoperability between vehicle and implement offered by Isobus, Autosar makes the development of software for implements more efficient. Software can be reused for different implements. In the presented solution, customer functionalities are decoupled from Isobus Application Services and from the Basic Software on the control units of implements. This improves the integration of functionalities into implements.

The development of Basic Software can be approached separately. The Basic Software concept allows abstractions from electronic control units. In this way, hardware changes can be handled without impacting the Application Software. Parts of Basic Software can be reused, which in turn leads to a higher software quality. Moreover, the presented solution is developed according to the standard ISO 25119 for Functional Safety.

**SUMMARY**

With Autosar, the automotive industry has established a standard for software architecture and methodology. Autosar counters increasing complexity in software development with a focus on the main market-relevant factor: The Application Software which contains the customer functionality. Even if the migration towards the standard is challenging, it enables more efficient development of software with reuse scenarios. Through expansion of Autosar with the network standard Isobus, the presented solution makes the standard accessible for development of agricultural electronics.

The expansion of Autosar via CDDs is generic and provides options for further domain-specific extensions. The presented solution, however, bridges the major gap for the Autosar migration in agricultural machinery. Its main advantages are the efficient integration of functionalities into agricultural vehicles, the reuse of functional dependencies in the Basic Software, the reuse of Basic Software parts and higher quality of software. Moreover, the standard supports Functional Safety, Security and Multicore electronic aspects [3]. The Autosar consortium guarantees the evolution of the standard with a large community and diverse competencies. Last but not least, agricultural machinery players can contribute to the consortium for domain-specific extensions.

**REFERENCES**