Artop AUTOSAR Tool Platform

Artop Whitepaper

Updated: February 2010

Michael Rudorfer[1], Stefan Voget[2], Stephan Eberle[3]

[1] BMW Car IT, michael.rudorfer (at) bmw-carit.de
[2] Continental, stefan.voget (at) conti-engineering.com
[3] Geensys, stephan.eberle (at) geensys.com

1 Motivation

The AUTOSAR development partnership has developed a standard for automotive system design over the last years and is becoming heavily used in automotive projects by now. AUTOSAR defines a complex methodology for several process steps in system design. These steps include configuration of basis software, high-level modeling of a software structure, research topics like the system generator and others.

Each step in this methodology has to be supported by dedicated tools. Today several tools from different vendors are available in the market, but there is no continuous tool chain that supports all of these steps. Tool vendors try to cover as much as possible, but as different expertise is needed for these steps and it can be doubted if one tool supplier can provide all parts of such a tool chain in an equally good quality. The goal must be to have optimized dedicated tools for each step in the AUTOSAR methodology with good and easy interoperability.

From a user's point of view it would be desirable to have an interoperable tool chain that is capable of being put together by modules from different vendors. In that way one has the chance to pick the best parts to support each step of the AUTOSAR methodology from the vendor that has his expertise in this special area. An open, extensible tool chain would also allow for OEM specific extensions like dealing with legacy software or connectivity to backend systems, which will never become standardized in AUTOSAR as they are different in every company.

Another fact is that the AUTOSAR standard is evolving constantly resulting in new versions of the AUTOSAR metamodel. Today there is a significant gap between the release of the specifications and the availability of tools, which support that release. A speedup would be highly appreciated.

AUTOSAR defined an XML based exchange format for a specific part of the V-Cycle, i.e. for the system and BSW configuration. This facilitates the definition of tool interfaces but does not define them in depth. E.g. XML is a file format, but not an in-memory storage format. As the AUTOSAR methodology represents transformations on an XML file level each tool needs a basic set of features that realizes the handling of these files and the transformations into an in-memory storage format. The value of a tool is mainly in the functionality that is user visible, but not in these background features.

Nevertheless, the handling of several tools that interact on the files stored in the AUTOSAR exchange format is, at least in theory, possible. But, moving data from one tool

via the exchange format into another tool means, that it is transferred from a proprietary in-memory storage format into a standardized exchange format and again into another proprietary in-memory storage format. This is slow, has the risk of information loss and is error-prone. In addition it makes the traceability on object level difficult and error messages cannot be referenced between the tools. Therefore, it is still a challenge to transfer information between independent tools.

The ideal implementation of the tool chain includes lossless tool interfaces and well documented interactions between the tools. This can be reached by providing a common model framework that all user relevant features are based on. This can realize a harmonized user friendly environment where, e.g., all changes done in one tool part are visible to other tooling parts immediately.

The idea of Artop (AUTOSAR Tool Platform) is to establish a new approach that makes it possible to jointly develop a basis for an integrated tool chain. Its key idea is based on the observation, that there are basic functionalities, which every AUTOSAR tool needs. No tool vendor can differentiate himself from his competitors by providing these core features. They just have to be implemented for any AUTOSAR tool. Although this is not a challenging task, this work has to be done. Instead of implementing this base functionality required for every AUTOSAR tool at every company independently we propose to just join forces and implement this core in an open source like way. That means, that this basis is implemented by a handful of companies together inside of the Artop User Group and the resulting tool base is commonly shared with all members and partners of the AUTOSAR development cooperation.

This helps to establish a common extendable basis for AUTOSAR tools that improves tool quality, tool interoperability and development speed.

2 Artop User Group

To enable such an approach as stated in chapter 0 some companies with a strong interest in AUTOSAR tools have created the Artop User Group. Artop is the acronym for “AUTOSAR Tool Platform” and has the goal to create a platform that serves as a basis for creating dedicated AUTOSAR tools. The platform only contains non-competitive basic parts of an AUTOSAR tool and it is a jointly developed infrastructure for AUTOSAR tools.

The platform shall allow for faster creation of dedicated commercial tools by freeing the tool vendor from re-implementing tool basics that are required for any AUTOSAR tool.

The resulting platform is not a replacement for commercial tools. The goal of Artop is not to provide a free-of-cost tool to the end-user, but to provide a tool platform to all AUTOSAR members and partners that allows for development of better products in less time with improved interoperability and quality.

The focus and content of the Artop platform is designed and initiated by the so called **Design Members**, currently the companies Continental, Geensys, PSA and BMW Car IT. They all contribute to the platform in one way or another and define the strategic direction of Artop.

Artop is open to any AUTOSAR member or partner that is interested in a jointly developed tool infrastructure and is willing to contribute to such an activity. They can join the Artop User Group as **Contributing Members** and participate in the development of

the platform. This can be in the form of extending and maintaining existing parts of Artop or by adding new sub-projects to the platform, for which they take the responsibility.

A third way to take advantage of Artop is to just use it and build a end-user tool or product on top of it. Those companies are called **Adopters**. They do not directly contribute to the Artop base, but use the provided platform to enable their tool development. Of course an Adopter indirectly contributes by providing feedback to the User Group. By this, an ecosystem around Artop can be established which pushes forward the whole initiative.

3 Scope

Artop is an infrastructure platform implementation of common base functionality for development tools that are used to design and configure AUTOSAR compliant systems and ECU's. For this Artop provides an API to the AUTOSAR metamodel that enables:

- The creation of commercial AUTOSAR tools with shared base functionality.
- The availability of AUTOSAR tools close to the point in time when AUTOSAR specifications are released.
- Early feedback to the AUTOSAR specification development process.
- Improvements in the cooperation of AUTOSAR members and partners in their development tasks in the various working groups.
- More competition among tool vendors in their area of expertise.
- An increased level of interoperability and integration of AUTOSAR tools.

This avoids redundant development of the non-competitive infrastructure platform by different tool vendors and supports a development process that allows for well-tested tools for all steps of the AUTOSAR methodology.

Artop provides basic building blocks which are typically required for realizing AUTOSAR-related tools. It encompasses implementations of AUTOSAR metamodel releases and a number of related services including AUTOSAR XSD conforming serialization, rule-based validation, tree and form-based views and editing, and template-based target code, documentation and report generation, and more.

Artop is based on top of the Eclipse Platform and other proven Eclipse technologies, such as the Eclipse Modeling Framework (EMF) [3], Since the available Eclipse technologies sometimes have to be adopted to be used in Artop, with respect to the given context, Artop provides also a set of complementary extensions to the different Eclipse projects. They simplify the implementation of above mentioned building blocks of Artop and may also be useful for realizing AUTOSAR tools on top of it.

Artop finally includes a set of examples showing how the provided basic building blocks and Eclipse extensions can be used in practice.

4 Architecture Overview

As mentioned in the previous chapters, the Artop platform shall be the basis to build commercial AUTOSAR tools on top of it. A typical product that makes use of Artop would typically consist of four layers. Only the middle two layers are within the scope of Artop

(see Fig. 1).

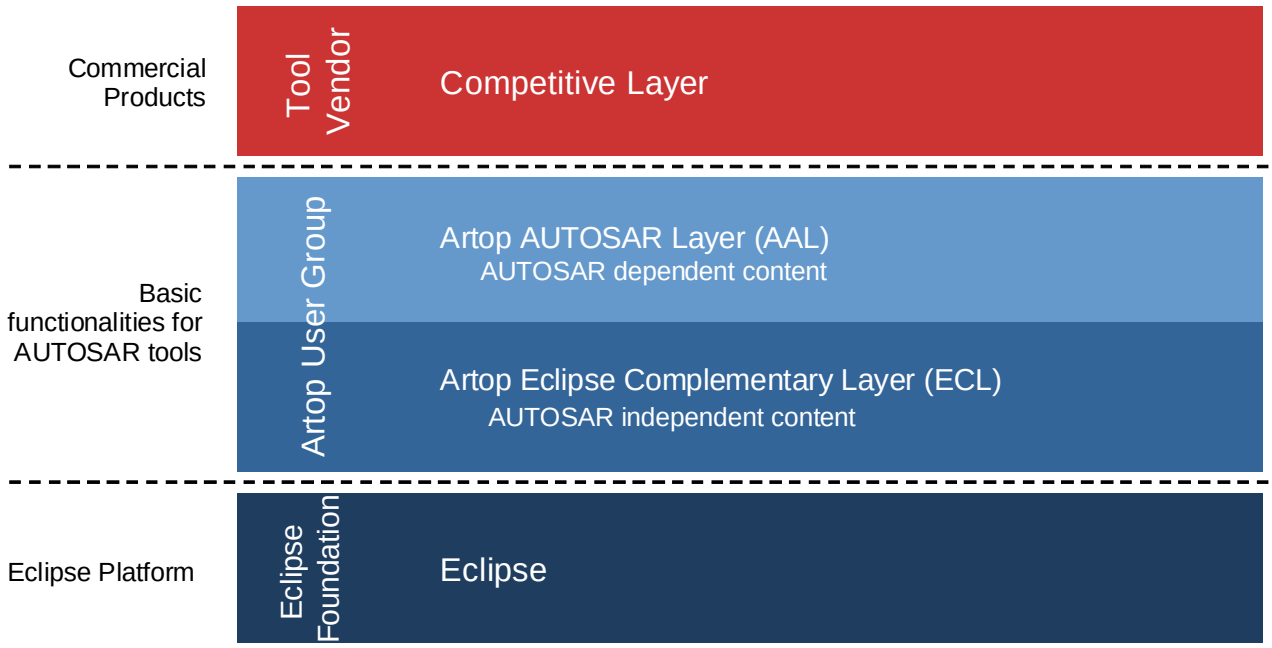

Fig. 1 – Artop Layers

The Artop platform itself is developed on top of **Eclipse**. Eclipse is an open source project that provides a platform for the development of software applications [2]. It already provides a lot of frameworks and functionality that can be used for creating domain-specific development tools. The Eclipse platform is the lowest layer in the Artop architecture, but is of course not developed inside the Artop User Group.

On top of Eclipse the two Artop layers are located. Artop shall provide basic functionality, which is useful to anyone who wants to create AUTOSAR tools. The contents of the Artop layers are meant to contain the non-competitive parts of such a tool. Functionalities like the implementation of the AUTOSAR model, parsing of models, simple editing capabilities and so on are required for every AUTOSAR tool.

The difference between the two layers is that the upper layer, the **Artop AUTOSAR layer (AAL)**, contains code that makes use of AUTOSAR specifications. This code contains intellectual property that is owned by AUTOSAR so this code can only be shared with companies that are AUTOSAR members or partners.

The lower layer, the **Eclipse Complementary Layer (ECL)** contains code that is needed for Artop based tools and is also developed by the Artop User Group, but has no AUTOSAR specific parts, so it could in theory also be shared with non-AUTOSAR members. It is planned to reduce the Eclipse Complementary Layer as much as possible and to migrate the contents of this layer to Eclipse itself. That means to contribute it directly to Eclipse.org and so be able to jointly work on those parts in an even larger community, which will reduce the direct effort of maintenance for the Artop User Group.

On top of Artop there will be a layer of special plug-ins with competitive content, the **Competitive Layer**. Any sophisticated functionality that a company wants to provide as a commercial product goes in this layer. As these plug-ins are the extensions of the platform where the tool vendor differentiates himself from his competitors, of course these plug-ins are not developed inside the Artop User Group, but are regularly developed by the tool vendor.

An important aspect to note is that the Artop licenses and the Eclipse Public License (EPL) allow that all four layers can be assembled together and be sold as an individual commercial product.

5 Components

Artop's first release candidate was available end of October 2008. Since then several versions of Artop have been released, the last one being Artop 2.0 that was completed in February 2010.. Artop 2.0 for the first time included support for AUTOSAR 4.0 just after AUTOSAR 4.0 was released itself. This chapter gives a brief overview of the software components that are part of Artop.

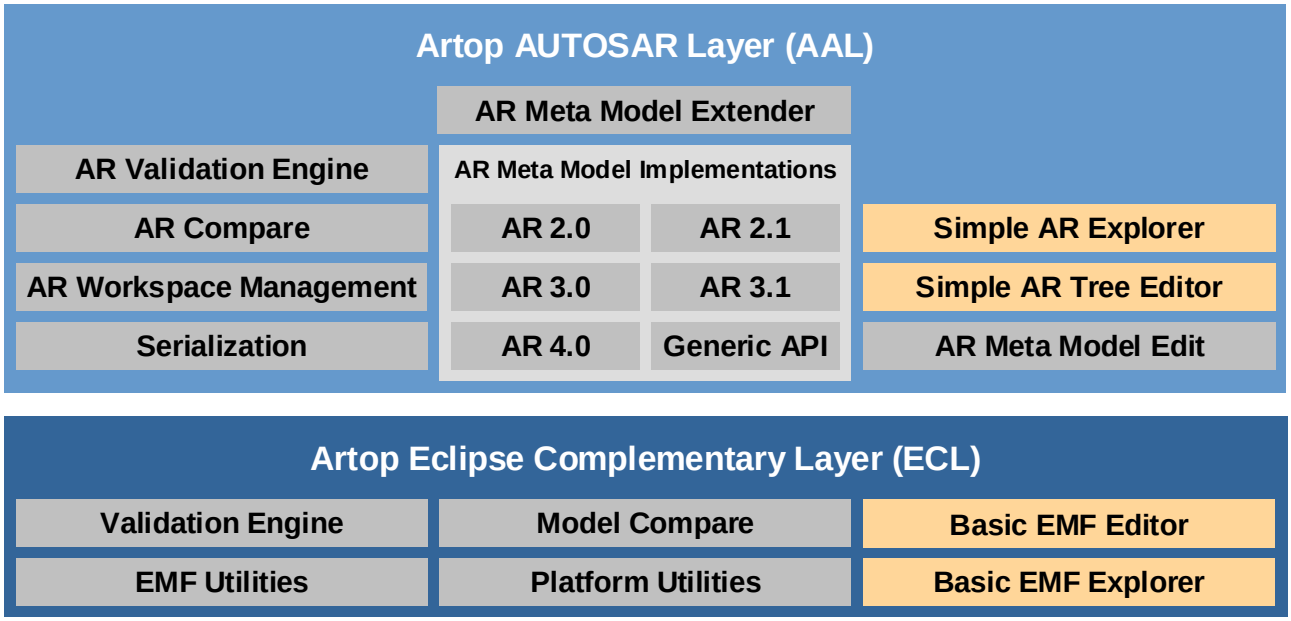

Fig. 2 - Artop Components

5.1 Main Components

The following components are the main components of Artop (grey components in picture).

5.1.1 AUTOSAR Metamodel Implementations

The most important part of the Artop platform is an implementation of the AUTOSAR metamodel.

The metamodel is implemented using the Java programming language and the Eclipse Modeling Framework (EMF). Currently Artop contains support for AUTOSAR release 2.0, 2.1, 3.0, 3.1 as well as AUTOSAR release 4.0. Support for upcoming AUTOSAR metamodel releases will be added with later versions of Artop. The goal is to provide metamodel implementations in Artop close to the point in time when AUTOSAR releases them.

Artop also provides a Generic API Metamodel API that allows to develop plug-ins that process or manipulate AUTOSAR models of different AUTOSAR releases with one common implementation. The Generic API is provided for certain commonly used areas of the metamodel. Naturally the Generic API cannot cover the full metamodels.

5.1.2 Serialization

The Serialization component provides file-based persistence for AUTOSAR models. It allows for serializing and de-serializing AUTOSAR models to and from AUTOSAR XML files, based on the XSD schema defined by AUTOSAR.

5.1.3 Workspace Management

The AUTOSAR Workspace Management supports managing of AUTOSAR models, which are spread over more than one AUTOSAR XML file. It integrates into the Eclipse workspace and provides the capability to load and save various file formats.

5.1.4 Metamodel Edit

Metamodel Edit helps to create user interfaces on top of the AUTOSAR model by enabling in-memory AUTOSAR models to be displayed and modified. It is based on the EMF.Edit framework and provides key features like content and label provider classes, property source classes and a non-transactional command framework to create, modify and delete model elements.

5.1.5 Validation Engine

Artop contains a generic validation engine that minimizes the effort to create validation constraints for AUTOSAR models. This component can validate constraints that correspond to the standard AUTOSAR constraints, as well as project specific or custom constraints that can easily be added via the extension point mechanism.

5.1.6 Compare

Artop provides functionality that allows to compare AUTOSAR models. This allows to easily track changes between different version of an AUTOSAR model file. The functionality is based on EMF Compare that additionally allows to visualize the changes between models.

5.1.7 EMF Utilities

The EMF Utilities provide a number of useful EMF extensions and helper classes. They can and should be used for simplifying and unifying the implementation recurrent EMF-related tasks.

5.1.8 Platform Utilities

The Platform Utilities provide a number of useful Eclipse extensions and helper classes. They can and should be used for simplifying and unifying the implementation recurrent Eclipse-related tasks.

5.2 Example Components

Some example components, mainly UI components are also provided with Artop (orange components). These components show the usage of Artop, but are not meant to be used by an end-user. They will not be fully tested and documented and there is also no guarantee that the API of these components will be stable.

5.3 Future Components

There are more components that shall in future be developed as part of Artop but that are not contained in the current release. The Artop Design Members will decide on the content of the platform. Potential contributors are encouraged to make proposals for more

components that shall be part of Artop. Any meaningful extension that is aligned with the purpose of Artop is highly welcome.

6 Release Planning

AUTOSAR has released several versions of the metamodel so far. As it is the goal of Artop to support the AUTOSAR partners and members by providing a tool framework early in time, the release planning of Artop is in-line with the release planning of AUTOSAR. For each new metamodel version of AUTOSAR a corresponding release of Artop will be planned in time, i.e. in optimal case at the same date, but at least close to it. Such a release will include the support of the new AUTOSAR metamodel release as well as the former ones.

7 Mode of Cooperation

As it is the goal of Artop to develop the non-competitive parts of AUTOSAR tools the contribution of all AUTOSAR partners or members is welcome. Started with a small development group the collaboration of the participating companies had been organized from the beginning on with respect to the goal to incorporate more and more members. At the end such a development of a basic tool platform can only work if there is an active, growing community that participates in Artop.

Technically the distributed development uses similar tools like the AUTOSAR partnership, e.g., Subversion for version management, Bugzilla for change management, and a Wiki for information exchange. The similarities to the AUTOSAR partnership help to cooperate with an environment that is well known to all AUTOSAR members and partners for years.

But the development would not run without the dedication of the partners within the Artop User Group. The common vision of the needs and advantages of the tool platform drives the cooperation.

8 Outlook

The development of Artop is driven by the reactions and interest from AUTOSAR members and partners.

- How are OEMs, Tier1s using Artop?
- How are tool vendors using Artop as a base framework for their commercial tool development?
- Will further AUTOSAR partners and members be interested in joining and supporting Artop?

The answers to these questions will mainly influence the success and the further development of Artop. The Design Members of Artop are open to adjust Artop to the needs and requests from other AUTOSAR partners and members. Only if Artop will be used a successful development can be continued.

By now there already several companies joined the Artop User Group as Contributing Members so there is a growing community of Artop users.

The members of the Artop User Group drive the development because they want to use it themselves. The company internal tool-chain strategies will be adjusted with respect to Artop to be able to profit from the advantages of a collaboratively developed platform.

9 References

[1] AUTOMOTIVE OPEN SYSTEM ARCHITECTURE http://www.autosar.org
[2] Eclipse, http://www.eclipse.org
[3] Eclipse Modeling Framework, http://www.eclipse.org/emf/
[4] Harald Heinecke, Michael Rudorfer, Paul Hoser, Christoph Ainhauser, Oliver Scheickl, Enabling of AUTOSAR system design using Eclipse-based tooling, In Proc. Embedded Real-Time Software Congress (ERTS), Toulouse, France, 2008