Eclipse based architecture of the EDONA platform for automotive system development

F. Ougier[1], F. Terrier[2]

1: Renault, 1 avenue du Golf, Guyancourt, F-78288, France – francois.ougier@renault.com
2: CEA LIST, Gif-sur-Yvette, F-91191, France – francois.terrier@cea.fr

Abstract: EDONA is a platform project started in 2007 in order to deliver the infrastructure for a seamless tool chain for the development of automotive embedded systems. The goal of this paper, a couple of months from the end of the project, is to explain how the initial goals have been extended, implemented, and secured by opening it and relying on external long-lived organizations.

Keywords: Automotive embedded systems, development platforms, AUTOSAR, open source sustainability

1. Introduction

Gathering at the national level the totality of the main and large industry actors of the field of automotive embedded software, EDONA is a project of the "System@tic Paris-Région" cluster. EDONA aims at facilitating the assembly of seamless tool chains for the development of automotive embedded systems. Modular, interoperable and adaptable to the various needs of the actors and trades of the car industry, it integrates from the start the requirement of compliance to automotive standards, with a particular focus on AUTOSAR [10].

The project is lead by Renault and the form chosen is (1) the creation of a technological reference platform and then (2) its specialization on business processes of the field (see [1] for an introduction to EDONA at its beginning). Each specialized development tool chain will result from the following sequence of steps:

- an expression of needs controlled by a main industry actor particularly interested by the technology followed by an activity of technology transfer jointly carried out by both research and industry
- a coherent integration allowing to support, by improving them, the existing work processes. Integration uses the platform and extends it with various modules provided by the laboratories and SMEs
- a full-scale experimentation carried out by at least one industrial company for each development tool chain.

The goal of this paper is, while going through the main results of the project, to describe and outline how the initial stakes naturally lead:

1. to open it to the outside world, and to start delegating its future evolution to external organizations;
2. to extend each experiment beyond the project plan in order to emphasize the value of the benefits brought by a sustained development of the approach.

2. Breaking down: architecture decomposition

2.1 Stakes

At the origins of EDONA was a strategic decision of the Automobile & Transportation group (a.k.a. Num@tec) of the System@tic Paris Région French R&D cluster to federate the projects they labelled. This goal would be achieved by specifying and building a number of platforms that would host all their tools and concept developments. EDONA was to be the first one, covering automotive electronic systems design.

This goal matched the expectations of some car manufacturers and suppliers who in some situations began to perceive the detrimental effects (in terms of schedule and cost as well as specification and implementation quality) of using disconnected tools along the V cycle (see figure 1 below).

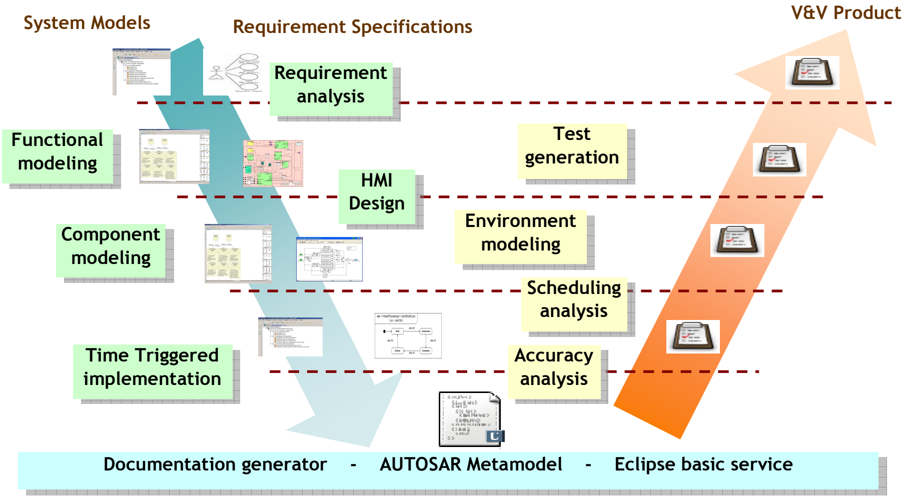

Figure 1: V-cycle areas addressed by Edona

This became really significant when the AUTOSAR consortium gained momentum. AUTOSAR uses interface descriptions in XML, diagrammatic representations, and an UML meta-model. Those elements make it very difficult to do anything without tools, and immediately raise the issue of connecting them to the requirement documents and to the various system engineering, modelling and testing tools that were more or less being experimented and deployed around 2006.

2.2 Constraints

Assembling a “real life” seamless chain means:

- Having a common technical infrastructure
- Being able to connect existing commercial tools
- Being able to plug innovative research tools
- Maintaining the infrastructure for a long period, letting innovation

Since, as figure 1 shows, a large number of tools are addressed, the cheapest solution is to use an existing, ubiquitous infrastructure, and to concentrate on developing the glue and the adapters to the tools.

2.3 Architectural principles

Therefore, to reduce integration efforts and harmonize modelling features and interaction mechanisms, EDONA is built upon Eclipse.

A set of basic Eclipse components are reused such as EMF model repository, ATL for model transformation, etc.

This is the layer where automotive dedicated technologies are integrated: namely, an AUTOSAR reference meta-model and various facilities such as multi-file storage of AUTOSAR models and model validation.

High level tools (either open source or commercial) are integrated within the layer corresponding to their use. They cover the whole development process (incl. requirement, design, coding and testing).

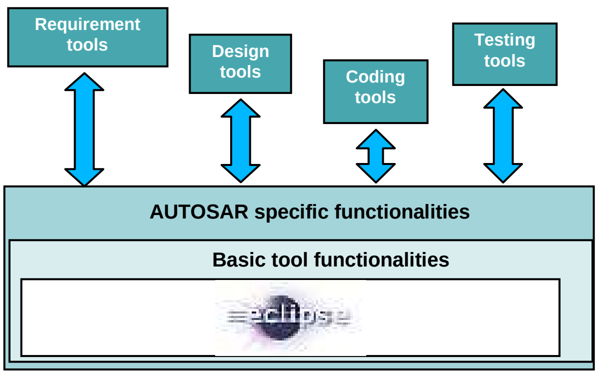

Figure 2: EDONA platform architecture (layers)

Seamless operation is made possible because the tool adapters are coupled with data models on the platform side. Those are then orchestrated using various technologies (from model transformation using ATL to more manual scripting mechanisms) selected depending on various feasibility criteria: complexity and stability of the model, real time round trip engineering required or not, etc…

This pragmatic approach was selected over more aesthetically gratifying solutions like a large pivot meta-model backed by some sort of application bus because it was far cheaper to develop, evolve, and maintain: seamless operation is not to be achieved between **any** tool, but only along small subsets which are contiguous in the life cycle.

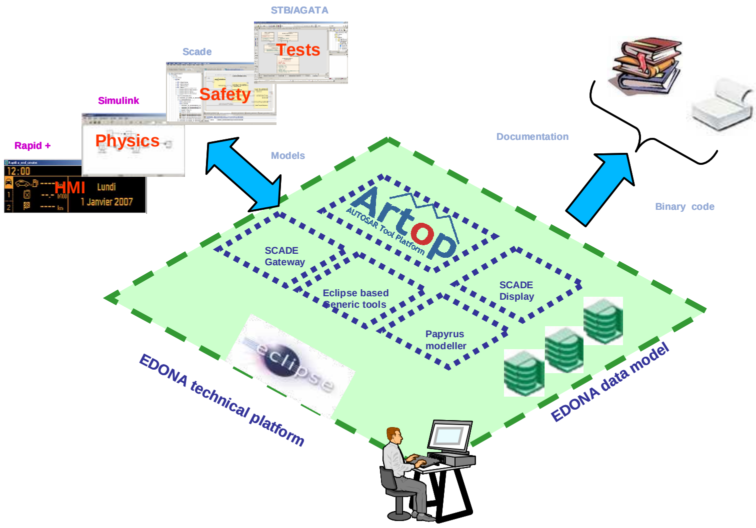

Figure 3: Edona platform architecture (floor plan)

2.5 Architectural trade-offs: an example

Nowadays, if one builds an automotive system using a model-based approach, the software is very likely to end up as a collection of AUTOSAR components.

Therefore, an interoperable platform like EDONA must be able to handle AUTOSAR objects. Before even thinking about generating them from higher level objects, this first means representing them in an Eclipse workspace as an instance of the AUTOSAR meta-model, and being at least able to edit them, save them, and check them against some set of rules.

These features represent an essential part of the EDONA work package 5 deliverables. They have been structured in sub layers, in such a way that features that really require an AUTOSAR license are strictly confined.

Other AUTOSAR members naturally had the same idea, and when they decided to setup what is called the Artop user group (where several EDONA members are actively involved) to put it together, the timelines were such that those EDONA outputs could be transferred to them.

They have been shipped as a part of the Artop 1.1 release, which is available since March 31, 2009. Details can be found at their site [7] (see figure 4).

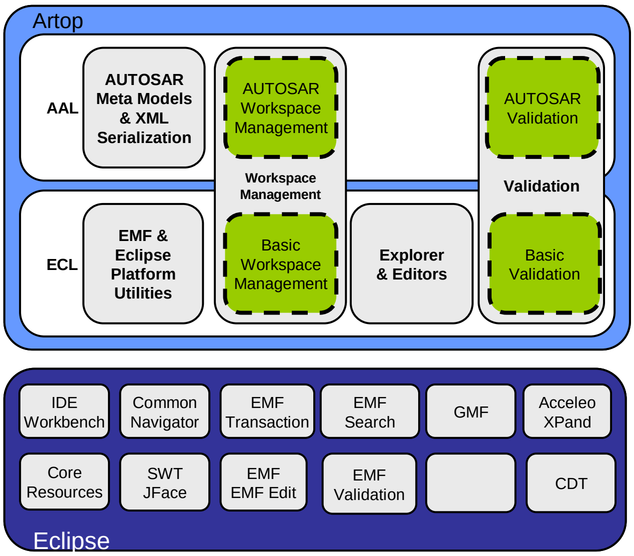

Figure 4: Autosar object management in Edona

This first architecture will evolve to create stronger separation among the general purpose components and the autosar adherent ones, in order to reduce the exploitation constraints on the non autosar specific component. Concretely, EDONA supports an Eclipser project proposal, Sphynx [11], in which all the ECL layer of Artop can be transferred and shared with a larger community.

3. Building up: application synthesis

3.1 Principles

The initial project plan, being based on vertical use cases that would activate platform elements, included, as outputs of the development and validation phases, a set of demonstration setups. We shall see in chapter 4 that the project team decided to extend them into full-fledged demos.

The next 4 paragraphs give an outline of what came out of those use cases.

3.2 Implementing and tracing requirements

The first vertical use case, used as a guideline for EDONA work package 1, is component-based modelling. Continuity of component based development processes is ensured thanks to the integration of three elements (see figure 5):

(1) Functional architecture modelling and refinement with EAST-ADL 2 [2] and the Papyrus Eclipse modeller [9];
(2) AUTOSAR configuration using the Artop meta-model;
(3) Translation of EAST-ADL 2 component architecture into AUTOSAR descriptions with the ARGateway.

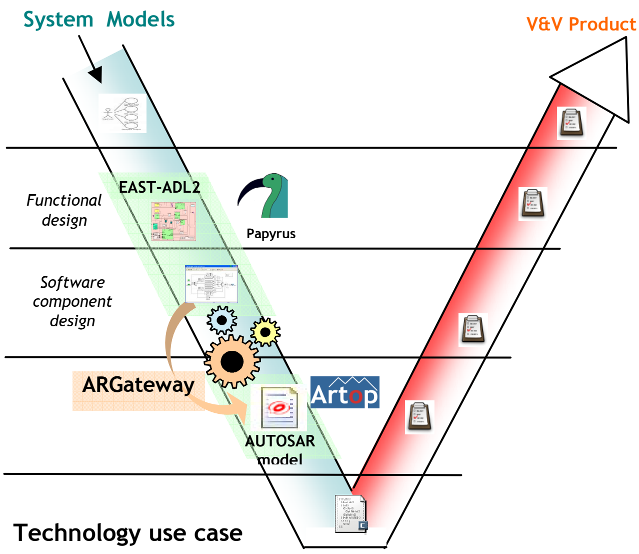

Figure 5: Model transformation

They provide a first tool chain integration from high level modelling down to AUTOSAR software component description, ready for code generation and deployment. It is ensured by:

- conceptual alignment of the lowest level of EAST-ADL 2 modelling with AUTOSAR meta-model and their implementation in Eclipse components, respectively, in the Papyrus UML modeller and in Artop basics.
- model transformation between EAST-ADL 2 descriptions and AUTOSAR.

3.3 Robustifying

Within the EDONA project, work package 2 is focused on safety developments. To reach this general goal, the project is based on the PharOS technology [3]: a consortium has been built to deploy it industrially.

Technical details:

PharOS technology enables design and implementation of embedded real-time multitasking and time triggered systems. The PharOS system is designed to be "correct by construction" and to have a deterministic behaviour (i.e. predictable and reproducible behaviour in both temporal and logical domains) which is reached by mastering all sources of asynchronism (i.e. pre-emption due to scheduling policy, variations of the execution times, and communication delays) [4].

The execution support (from the hardware system architecture to the software kernel) is totally abstracted and relies on a rigorous execution control concept: tools are then needed to safely swiftly go from maintainable system specifications to the PharOS binary.

EDONA helps developing 2 significant tools to extend PharOS suite:

- A tool box running on Simulink to design PharOS applications coupled with a code generator. The first element abstract PharOS' language and adds precise constraints on software architecture (tasks, communication) and real time behaviour (timing parameters). Then the code generator produces, in an integrated and continuous flow, the source code associated to the designed application. With the system view of Simulink, this tool makes the design of PharOS application easier and faster. With the code generator, engineering activity is focused on the definition of the software and no longer on implementation. Repeatability of the generation reduces detailed design effort and coding errors.
- A simulation tool running on host machine. It allows running embeddable code (including operating system and drivers) with simulation of time and environment. Furthermore, engineers can stimulate the designed software and get software responses and temporal behaviours without being constrained by real time and real environment. All these results can be analyzed off line, post simulation.

Combining both tools allows a complementary, incremental and 'in the loop' method as shown in figure 6 below:

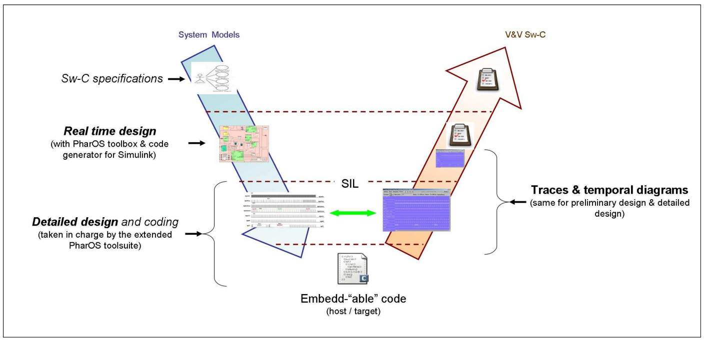

Figure 6: PharOS design in the loop

3.4 Testing

A third use case, deployed in and from EDONA uses formal techniques for validation of Simulink® functional algorithms. It basically consists of a Simulink® test case generation tool chain: integrated in the software development process, it helps designers improve the quality and robustness the models they design for software development cost and time reduction.

Once the model is developed, the designer can realize a validation & verification cycle on the model (see figure 7). The tool allows obtaining structural and functional test coverage: detection of deadlocks, data overflow, or unused algorithm parts, MC/DC ("Modified Condition/Decision Coverage", every condition and point of entry and exit in the program

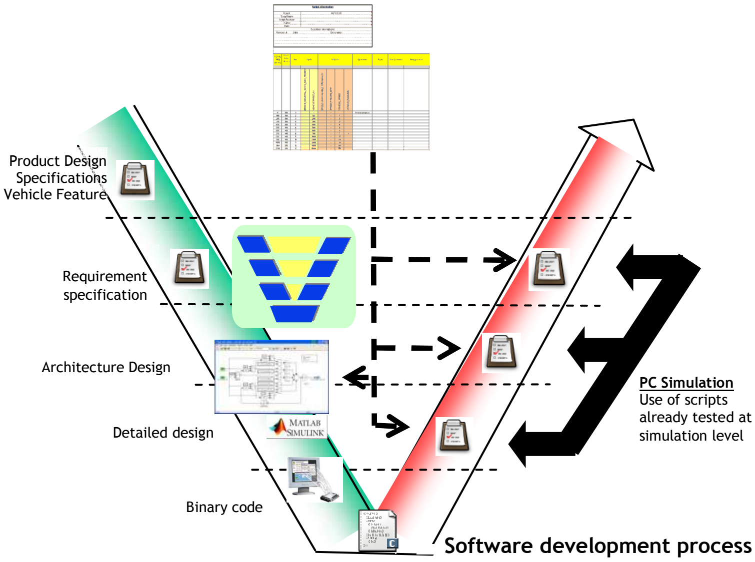

Figure 7: Testing loops

has been invoked at least once) or signal ranges.

Through a user-friendly interface in Safety Test Builder, user is able to work step by step for model testing. Safety Test Builder manages the interaction between user, test cases generation, Simulink models and export in Excel format. The export can be reused in other environments as, for example, for management of non regression tests or for automatic execution of multiple test sequences (see figure 8).

The generation of tests is based on symbolic automata simulation. Using formal techniques, it interprets the semantics of the model instead of just looking for numerical values and playing them on executable code or simulator. The purpose is to provide coverage improvement of the tests performed on the application.

The use of coverage criteria in conjunction with symbolic simulation reduces strongly the combinatory explosion when generating test cases (see [5]).

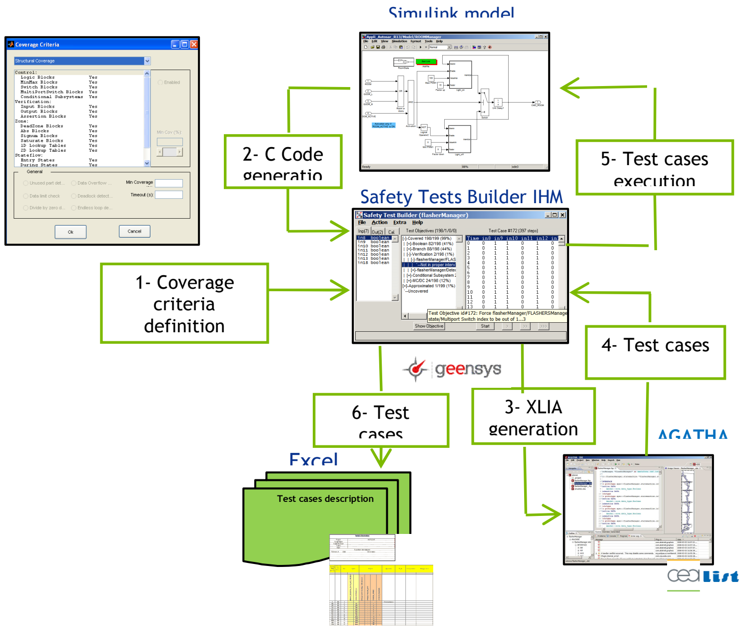

Figure 8: Testing process

3.5 Specifying HMI

A fourth use case, from EDONA work package 4 is to provide interoperable tool chains for Automotive HMI design. The issue is slightly different here: roughly speaking, OEM and supplier are supposed to use instances of the same tool class to execute two halves of a design operation (equivalent to a functional model and an implementation model), but the market situation is such that they are not likely to use the same instance. So the problem turns into the very classical one of finding a neutral format between similar but incompatible tools (see figure 9).

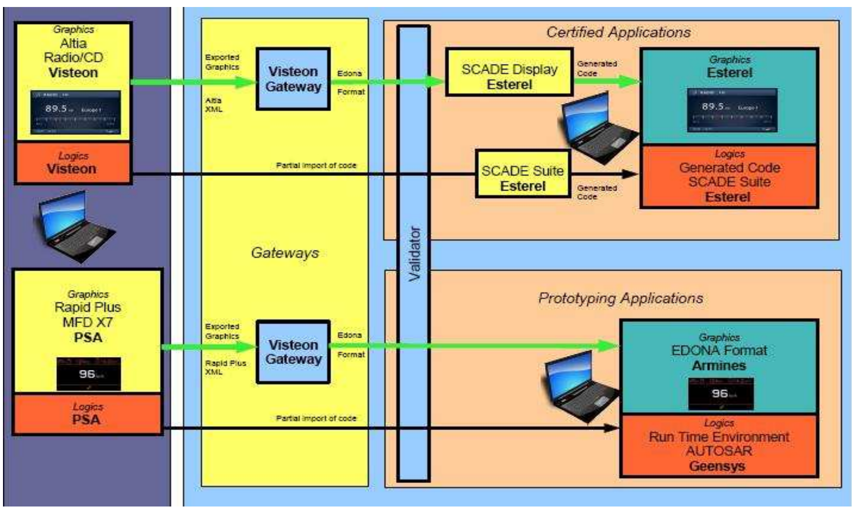

Figure 9: Converting HMI design files from Altia and Rapid plus to the EDONA format

Interoperability between various HMI design tools is a major concern when it comes to reducing development times. When a car manufacturer uses a given design tool to make an interactive HMI specification, suppliers often have to re-start from scratch since they use different software (see figure). That is exactly the issue addressed by the EDONA HMI format. Based upon the SVG Tiny 1.1 specifications, the EDONA HMI format has been extended to enable most HMI graphical needs, including basic logic handling (aka micro functional layer) [6].

Once specified, the EDONA HMI exchange format has been implemented through different gateways ensuring export/import capabilities by HMI design tools. Thus, graphical aspects of a design can now be preserved and exported into any other EDONA compliant tool.

In addition, Eclipse-based viewer and tools have been implemented by work package partners to handle the EDONA HMI format. This format makes it possible to seamlessly integrate functional mock-ups in the AUTOSAR-compliant system they belong to, in order to review the HMI design in action or simulate it while debugging its sibling software components.

4. Disseminating: component sustainability

The EDONA project has adopted a technical approach strongly based on pushing open standards of the domain that are designed to ease technology and tools evolutivity. This can be illustrated through several typical use cases in EDONA.

AUTOSAR: even if its use is restricted to the community having the rights to exploit it, it fulfils the objectives of openness and evolution by providing common approaches to provide flexible integration, reuse and evolution of components. Moreover, the consortium ensures a strong activity of maintenance and extension of the standard along several years with short update cycles. From EDONA point of view, exploitation of this standard has to be facilitated by providing tool development enablers open to the whole community. This is concretized by the distribution of the Artop platform implementing the AUTOSAR meta-model upon open source technologies, i.e. Eclipse platform and components. Artop itself is provided as "AUTOSAR common source" (i.e. open source for entities having the right to exploit AUTOSAR specifications).

UML profiles, even if UML and the profile mechanisms remains largely hidden for EDONA users, the technical approach chosen for providing domain specific languages and authoring tools (modellers) is totally based on the openness and extendibility mechanisms of UML and its profiles. This has allowed providing a definition of the EAST-ADL 2 language sharing a large set of concepts with formalisms dedicated to similar conceptual activities, namely, with SysML and MARTE. This has permitted for example, an easy integration in EDONA of plug-ins for scheduling analysis of AUTOSAR architectures developed on top of MARTE in the INTERESTED project. In the future, this will also ease integration of results from the IMOFIS project, allowing safety analysis based on SysML and MARTE modelling. Technically the dedicated modellers themselves are also implemented on open technologies and mechanisms provided by the Eclipse Papyrus modeller. In this case, dissemination of the modellers, and language definitions (at least for SysML, MARTE and EAST-ADL2) is made as open source results in the Eclipse foundation.

Even for dedicated technologies like STB/Agatha test generator, the principle of openness and evolution has been used: the technical kernel for test generation is built around a multi-semantics pivot formalisms (internal and hidden to the users) that allows using the same execution engine upon various formalisms. In EDONA it is used upon Simulink model, but the same kernel is also used in HeCoSim project upon Statemate models and will be extended in VERDE project for SysML models. There, potential developers community around the kernel itself, has still to be identified and developed, this is why open source approach is not chosen yet.

5. Lessons Learned and Perspectives

Technically, the EDONA platform has to allow integration and interoperation among a very large and diverse set of technologies and tools. Pragmatically, several initiatives of development of open and interoperable tooling technologies have been launched, namely, in Europe but also worldwide. So the strategy adopted by EDONA is to push each component in the context where it can find the larger set of users and contributors (Artop consortium, Eclipse Automotive project, Eclipse MDT project, etc.). However to ensure a constant and consistent capacity to integrate the next versions of the externalized technologies, EDONA partners, and more widely the Num@tec Automotive members, have launched a new umbrella project, called DESTAR. This project aims to animate a large community around AUTOSAR technology adoption. In this context, we foresee to set up a steering committee that will follow the evolution of the EDONA technologies and will coordinate integration and tuning actions.

7. Acknowledgement

The EDONA project, labelled by the Num@tec Automotive working group of the System@tic Paris Région Cluster, is supported by the "Direction Générale de la Compétitivité, de l'Industrie et des Services (DGCIS)" of the French government, the "Conseil Régional d' Île de France", the "Conseil Général des Yvelines", the "Conseil Général du Val d'Oise" and the "Conseil Général des Hauts de Seine".

This paper presents the status of the work and perpetuation approach adopted by the whole project consortium (who cannot be cited extensively due to the necessary limited size of a paper; for more information see [8]) and, particularly, by the project work package leaders: B. Sanchez (Continental), A. Tournadre (Delphi), P. Le Corre (Johnson Controls), H. Dufau (Visteon) and A. Loyer (PSA**).**

8. References

[1] F. Ougier & F. Terrier: “EDONA: an Open Integration Platform for Automotive Systems Development Tools” In proc. 4th European Congress Embedded Real Time Software (ERTS), Toulouse, France, 2008

[2] Cuenot et al.: “Managing Complexity of Automotive Electronics Using the EAST-ADL” In proc. 1st IEEE-SEE International Workshop UML and AADL, Auckland, New-Zealand, 2007

[3] David, V. &al., “Safety properties ensured by the OASIS model for safety critical real time systems”, IFIP-SAFECOMP’98 Conf., Germany, 1998

[4] Aussaguès, C., Chabrol, D., David, V., Roux, D., Willey, N., Tournadre, A., "An OS for Multicore Embedded Systems Compliant with Automotive Safety Standards", Proc. of the Int. Automotive Electronics Congress, Embedded systems and Standardisation Session, Paris, France, 2009

[5] C. Gaston, P. le Gall, N. Rapin, A. Touil: “Symbolic Execution Techniques for Test Purpose Definition”, In TestCom06, 18th IFIP International Conference on Testing Communicating Systems, New York, 2006

[6] Temmos et al.: “Automotive User Interfaces”, 6th International Conference on Scalable Vector Graphics, Nuremberg, Germany, 2008

[7] www.artop.org [8] www.edona.fr

[9] www.papyrus.org; http://wiki.eclipse.org/MDT/Papyrus

[10] www.autosar.org [11] www.eclipse.org/Sphinx