

# Towards Enabling Cross-Organizational Modeling in Automotive Ecosystems

Eric Knauss<sup>1</sup> and Daniela Damian<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering  
Chalmers | University of Gothenburg, Sweden  
eric.knauss@cse.gu.se

<sup>2</sup> Department of Computer Science  
University of Victoria, Victoria BC, Canada  
danielad@uvic.ca

**Abstract.** Automotive engineering is characterized by relying heavily on complex supplier networks as well as by strong dependence from hardware and software component development. OEMs (original equipment manufacturers) coordinate and integrate the work of hardware and software component suppliers and to an increasing amount develop application software themselves (component suppliers can be internal). For OEMs the transition to model-driven development promises potential reduction in the development lead-time of complex in-vehicle automotive features, such as semi-autonomous driving, but it is not without challenges. For example, the verification of such features is still performed using mainly physical properties such as hardware benches and mule vehicles. While this step is necessary, it is not sufficient, because it does not allow early verification of design decisions to the required extent. In addition, the development speed of hardware and software components is (a) limited by hardware development cycles as well as (b) slowed down by unsynchronized software development cycles of key suppliers. This prevents detailed information from being available early and potentially resulting in expensive and late changes. Understanding this situation as an ecosystem of cross-organizational collaborations allows us to reason about challenges and opportunities of the interaction between the OEM and different component- as well as tool-providers. In this paper, we report first results from an exploratory study that involved interviews with one of our industrial partners, General Motors (GM). First, we describe our understanding of the automotive ecosystem. Second, we explore interactions and roles of different ecosystem actors based on workshops and interviews with engineers at GM.

**Keywords:** automotive, cross-organizational modelling, software ecosystem

## 1 Introduction

Automotive engineering is characterized by the complexity of the system under construction as well as the required supply chain. In this environment, development is market driven. If the market motivates development of a new feature,

the automaker (OEM, original equipment manufacturer) starts with high level design of the feature, maps it to the overall system architecture, and derives hardware and software requirements, which are most often given to suppliers for implementation. Once the suppliers deliver hardware and software components, the OEM starts integrating them into a final product. The complexity of design artifacts and supplier networks lead to the need to coordinate and verify design decisions across organizational borders.

With few exceptions, OEMs address feature development in a way that resembles the waterfall process, characterized by upfront requirements analysis, early design decisions with limited knowledge, and by being able to verify the success of the development only later in the process after integration of the components into a mule vehicle (a complete, often drivable vehicle with experimental and prototype components). This situation can lead to sub-optimal decisions, which often result in late changes and rework. Lean software development considers this to be waste and suggests to change the process in a way that allows to make design decisions later, based on knowledge about the performance of a number of candidate solutions [14]. In the automotive domain, this would require the ability to verify non-functional properties (e.g. task performance) of such candidate solutions very early.

In this paper, we report first results from an exploratory study in which we aim at understanding coordination needs between actors in the automotive value chain by interpreting it as an ecosystem. Our contributions are (i) a characterization of the automotive ecosystem based on related work in software ecosystems, (ii) a first sketch of crucial interactions and roles of different ecosystem actors based on workshops and interviews with engineers at GM, and (iii) a discussion of challenges and opportunities of changing the ecosystem in a way that supports to make *binding decisions* (decisions that would be expensive to reverse such as contractual or architectural decisions, e.g. about a certain microcontroller) later and to do early non-functional verification across organizational borders earlier.

## 2 Research Questions and Method

This paper provides a first step towards describing automotive engineering as an automotive ecosystem of interacting organizations and is based on our collaboration with engineers and technical leaders at GM based on the following research questions:

**RQ1:** Who are the actors and what relationships exist among actors in the automotive ecosystem?

**RQ2:** What are examples of challenges and opportunities that currently exist in the automotive ecosystem?

To answer our research questions, we start by characterizing and defining the scope of the automotive ecosystem based on related work in the field of software ecosystems. We then follow up with a qualitative case study based upon worked done at GM R&D to further our understanding of such an ecosystem. At this

stage, our research is exploratory in nature and the preliminary results we present here are based on aggregating discussions during workshops and unstructured interviews. In future research, we want to engage with representatives of potential internal and external actors in the automotive ecosystem in semi-structured interviews, with the goal to elicit a clear understanding of actor relationships and their coordination needs as well as measurable objectives to measure and continuously monitor success, health, and gains of the automotive ecosystem.

### **3 Roles and Relationships of Automotive Ecosystem Actors**

In this section, we explore relationships and interactions of actors in the automotive ecosystem based on literature on software ecosystem and interviews as well as workshops. We start with an example of how actors collaborate in automotive supplier networks and characterize this ecosystem based on related work in software ecosystems, before we report preliminary results from our ongoing interviews with practitioners on how these actors interact to create value. This is a first step towards a more formal assessment of the automotive ecosystem (e.g. based on [9]).

#### **3.1 Characterization of the Automotive Ecosystem**

In understanding ecosystems, one would draw on three fields in software engineering: open source software [15], modelling and architecture (e.g. software evolution, architecture, and product lines [2]), and managerial perspectives (e.g. business aspects and co-innovation [6]). Different strategies exist in software ecosystems, varying from widely proprietary ecosystems based on a semi-open partnership program to pure open source ecosystems [1,5] and literature discusses almost as many proprietary ecosystems as free-and-open-source ecosystems [12].

Previous research proposed [2] to characterize ecosystems based on their category (end-user programming, application, or operating system) and the scope of the ecosystem's operating environment (Desktop, Web, or Mobile). While we see application software and operating systems for embedded systems in the automotive ecosystem, it does not clearly fit into one of the suggested scope categories. Instead, we propose to see automotive components as cyber-physical systems to emphasize the increasing degree of connectivity between components.

Automotive engineering depends to a large degree on collaboration across a large supplier network, which generates a significant coordination need. It is characterized by the integration of different hardware and software components, thus it is touching on various levels in the ecosystem stack [4], including hardware, systems software, middleware, and application software. In a given electronic control unit (ECU) at least three components can be distinguished: (i) The hardware component, e.g. the microcontroller and peripherals, (ii) the middleware component, providing drivers, communication facilities, and other

enabling routines, and (iii) the application software component, which provides (parts) of functionality for user-facing features (e.g. semi-autonomous driving).

All of these components can be provided by different suppliers or be developed in house at the OEM, who is also responsible for integrating the different inputs. We would thus see the OEM in the role of an ecosystem coordinator and characterize it in terms of Jansen et al. [5] as privately owned and participation to be based on a list of screened actors. Actors of the ecosystem can be keystones (which have a huge impact on the ecosystem and provide room for niche players), niche players offering complementary services to what keystone players already provide, and the orchestrator/coordinator who is coordinating the ecosystem [6]. Those actors have various relationships among each other, including selling software, providing services, providing software and assets, endorse software, train consultants, and contribute to the artifacts produced by the ecosystem [13].

This integration requires to make binding decisions which typically, in automotive engineering, need to be made before the application problem is completely understood. Making such early binding decisions can lead to late changes (e.g. if the hardware is too weak to support a given algorithm), or to waste (e.g. if the hardware is more powerful and more expensive than required). Both problems can only be discovered late, during non-functional verification and validation through testing. Based on Farbey et al.'s classification of inter-organizational relationships, we classify the automotive ecosystem to operate on Rung 1 (market relationships with dominant focal firm) or, in some cases, where embryonic networking among actors occurs, as Rung 2 [3]. Other works by Manikas et al. and Schultis et al. about characterizing actor relationships in internal or non-FOSS (Free and Open Source Software) ecosystems [10,16] will allow a comparison of our data in future work. We assume that network analysis as proposed by Manikas et al. [10] will reveal more dependencies and interrelations of actors than in their case study because of the integrated development efforts.

Consider for example a case, where the OEM is developing the application software component in house. R&D Engineers would develop a new control algorithm, using Simulink for model-driven development and simulation to allow functional testing, before hardware is in place. System designers would then decompose the Simulink models and generate software from Simulink blocks. By using for example *software-in-the-loop* based functional verification, this development is independent from the ECU hardware development, which can be started in parallel as soon as the hardware requirements are sufficiently understood. However, non-functional verification on the integration or system level can only be done once the hardware is developed and the execution time of the application tasks on the target hardware can be measured.

### **3.2 Actors and their Relationships in the Automotive Ecosystem.**

To identify and discuss the coordination needs of the ecosystem actors, we analyze one typical scenario in the automotive domain that uses MDD (Model-Driven Development) to provide a situation where software can be developed

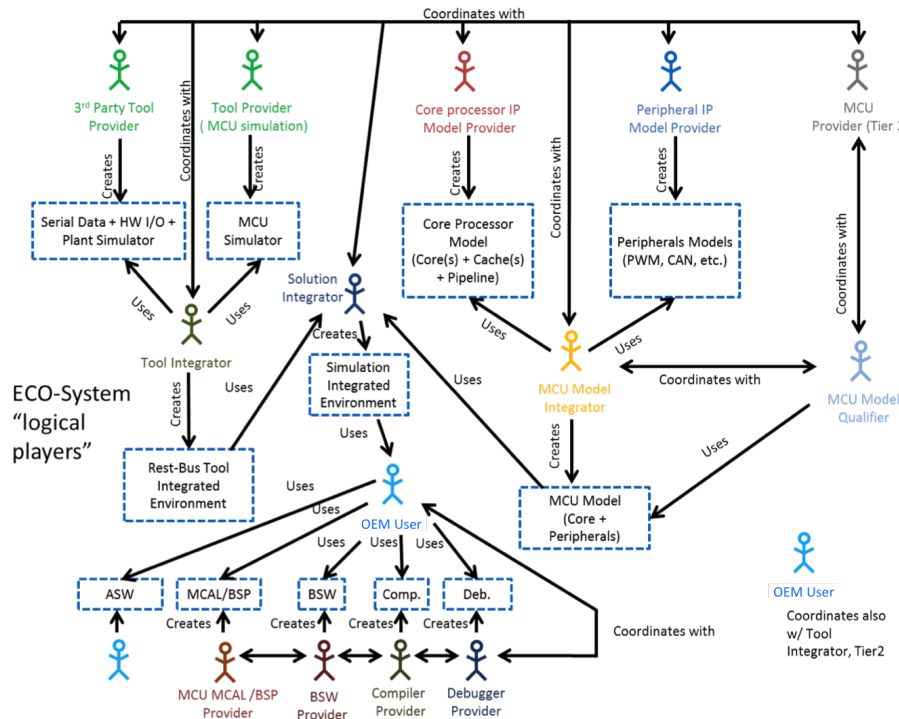


Fig. 1: Actors and relationships in a typical engineering example in the automotive ecosystem [17].

prior to the availability of MCU (Micro-Control Unit, part of the ECU) hardware. The same target code compiled for the actual MCU can be run on the simulated MCU at close to real time speeds and high degree of timing fidelity. Fig. 1 shows the actors and their relationships, as provided by one of our GM interviewees [17]. The figure emphasizes the strong coordination needs among actors and shows artifacts as well as services these actors exchange. The Figure shows logical roles of the actors, i.e. one organization could choose to fulfil several of the roles, e.g. both the model provider or the OEM user could also become the Model Qualifier for a given development project. We discuss these roles in the following.

The *3<sup>rd</sup> Party Tool Provider* contributes a model of the rest of the system, the plant model, which typically includes both a simulated controlled plant (e.g. an engine) or at least a simulation of the digital interface to it, as well as the simulated incoming messages from other MCUs in the system. This model allows the target code to interact with other parts of the system before these have been finished. Such models are usually exchanged using a third part tool format. For example, for an engine model in Simulink, a MDL file is exchanged, while for the network message model as Vector DBC file is included.

The *Tool Provider* contributes a standardised simulation environment (e.g. based on SystemC, a set of C++ classes that provide event-driven simulation) to (1) run the simulated MCU (2) execute the target code for the actual MCU on the simulated MCU.

The *Tool Integrator* combines both tools into the Rest-Bus Tool Integrated Environment, which allows OEM users to do Rest-Bus Simulation. Rest-Bus Simulation is a technique used to validate ECU functionality by simulating parts of an in-vehicle bus like the controller area network (CAN).

The *Core Processor IP Model Provider* provides a core processor model of the intended hardware (usually an MCU). The *Peripheral IP Model Provider* provides a model of the peripherals in which the core processor is embedded. The *MCU Provider* will provide the hardware (typically, this is done by an automotive Tier 2 supplier).

The *MCU Model Qualifier* ensures the requested level of accuracy of the model in representing the MCU hardware. Currently there is no formal definition of this role or default processes to qualify or certify the accuracy of models, yet this would be crucial to overcome challenges as discussed later in this paper.

The *MCU Model Integrator* uses Core Processor and Periphery Models to create a MCU Model. The *Solution integrator* integrates all models and tools into an integrated simulation environment.

The *OEM user* develops the application software component, which relies on the MCU hardware. It is important, that the development can start before the availability of hardware, but also that design decisions can be non-functionally verified early in the process. As development proceeds, uncertainty is reduced and more knowledge about constraints becomes available. Also, a higher accuracy of verification becomes necessary. While for later verification, a hardware board is crucial, the availability of a virtual board early can be an asset, if adequate accuracy is provided.

## 4 Opportunities and Challenges of GM's Automotive Ecosystem

Based upon the work done in GM R&D, our GM interviewees have stated that one of the biggest opportunities of the automotive ecosystem is to enable its actors to share accurate models of hardware, periphery, and middleware software *early and continuously* in order to facilitate later binding decisions and early non-functional verification. These models could then be partly refined as more knowledge becomes available. For this, development partners (e.g. OEM, Tier 1 and 2 suppliers) need synchronization points where they share their current level of knowledge. Examples of these include:

- OEM shares current version of Simulink prototypes and models with other actors in the ecosystem.
- Tier 1 regularly shares information about the task composition and the characterization of task execution time as it becomes available. This allows the OEM early simulation and non-functionally verification.

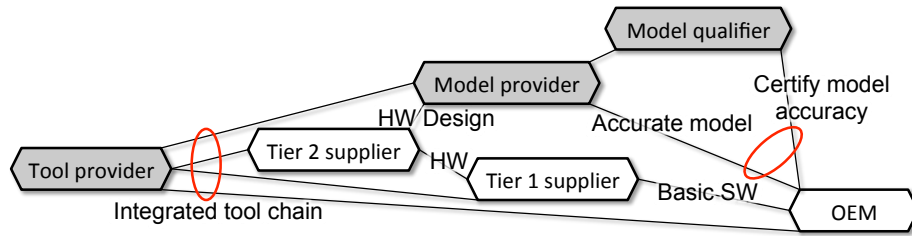


Fig. 2: Schematic view of abstract roles and their relationship in the automotive ecosystem. Actors to the left provide services and artifacts to actors on the right.

- Tier 2 shares information about the microcontroller design and its capabilities. This allows the Tier 1 supplier to estimate task execution times.

To increase the efficiency of this information exchange, our interviewees suggested to consider tool providers as part of the ecosystem, as they enable interoperability and exchange of relevant models. Also, while Tier 1 and 2 suppliers can be model providers, it could be preferable for some to rely on external modeling experts to create accurate models of their hardware in development.

From these observations, we extracted a more generic view on the automotive ecosystem (see Fig. 2). Basically, our study so far revealed two value chains relevant to automotive ecosystems. First, we identified the classic automotive value chains, where the OEM relies on delivery of HW and SW components by Tier 1 and 2 suppliers. Secondly, in order to support early non-functional testing (and consequently late design decisions based on accurate knowledge), a second value chain needs to be introduced to provide and certify accurate models of the HW in parallel to the main development stream (gray actors in Fig. 2). A Tier 2 Supplier might provide such models themselves or rely on an external Model Provider. Further, the accuracy of the models with respect of non-functional properties needs to be certified to create reliable models that allow testing *target-code-in-the-loop* (TCIL, a new simulation paradigm where application code compiled for the actual MCU runs on a simulated MCU).

**Opportunities in the Automotive Ecosystem.** The envisioned automotive ecosystem as sketched in Fig. 2 provides opportunities for win-win scenarios, because actors are not competitors. Tier 2 suppliers for example would gain a competitive advantage if they can more easily integrate into such an ecosystem, e.g. by collaborating with model providers (or by taking that role themselves) in order to provide models of their hardware early and update them regularly so that their accuracy continuously grows until the hardware is completely developed. Tool providers will benefit from a larger market of their integrated tools, which enable the TCIL cycle. Tier 1 suppliers would benefit from the ability to run regression tests on virtual boards quickly.

**Challenges in the Automotive Ecosystem.** As of today, OEM users may decide against sophisticated models of hardware as they can be more expensive than a hardware evaluation board, especially when considering that currently there is no formal and independent certification of the accuracy of models. Potential time-saving opportunities are typically not exploited because non-functional verification can only start when the evaluation board becomes available.

*Acceptance of modeling can depend on availability of certification processes.* Our interviews identify the lack of certification as one of the main technical challenges. A clear process for the qualification of models needs to be in place and best practices as well as standard collaboration models need to be defined. By this, the MCU Model Qualifier actor in Fig. 1 appears as one of the key roles in the ecosystem to allow for transparent and reliable assessment of model accuracy.

*Introduction of MDD might impact ecosystem health.* To enable a healthy ecosystem, it should be avoided that a keystone player becomes a dominator [11]. This could e.g. happen, if a specific Tier 2 supplier would be the only supplier that offers models with sufficient quality, thus becoming a monopolist in the example.

*Effective cross-organizational modeling depends on new solutions for legal concerns.* Providing accurate HW models to other ecosystem partners early may require Tier 2 suppliers to define provisions to protect against potential disclosure of their intellectual property. While such openness is required to leverage the opportunities our interviewees see in the automotive ecosystem, adequate licenses and contract models need to be introduced to offer sufficient protection.

*Cross-organizational modeling impacts local processes.* To decrease development lead-time by offering early non-functional testing as well as later binding decisions, actors may need to adjust their internal processes, such as those in sales and purchasing to include provisions that cover models of IPs in the contracts.

## 5 Discussion and Outlook on Future Work

In this paper, we analyzed the automotive value chain and supplier network as an ecosystem and explored the extent to which this allows us understanding actor relationships (e.g. information flows), challenges (e.g. coordination needs) and potential for optimization. An accurate charting of the automotive ecosystem landscape requires more interviews with technical leaders at OEMs, Tier 1 and Tier 2 suppliers, tool providers, and potential model providers and qualifiers. We report now on this ongoing work to gather feedback on our intended approach to understand the automotive ecosystem.

The ecosystem perspective offers a unique chance to analyze actors and their relationships in the ecosystem and to uncover challenges as well as opportunities, as shown in the example of the TCIL case. In previous work, we found that navigating the tradeoffs between *protecting intellectual properties*  $\longleftrightarrow$  *openness*



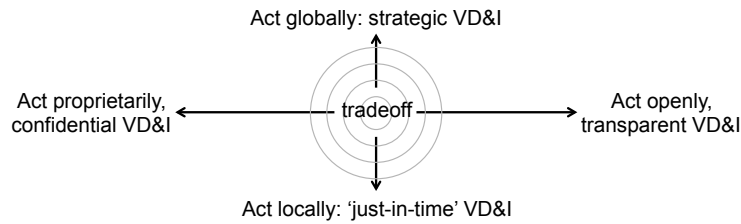


Fig. 3: Tradeoffs in software ecosystems, here with respect to VD&I (Virtual Development and Integration).

as well as between *global top-down approach*  $\longleftrightarrow$  *Responsive bottom-up approach* generates opportunities for shared understanding of requirements between actors in an ecosystem [8]. In Fig. 3 we represent such tradeoffs with respect to the Virtual Development and Integration process and challenges and impediments for leveraging the true potential of the automotive ecosystem. For example, an increased information exchange among actors in the ecosystem benefits the overall ecosystem, but requires that the intellectual property of partners is protected. Enabling actors to make their own design decisions and to share them with relevant other actors increases the responsiveness of the ecosystem as well as the potential to allocate resources where they are needed most, but still a coordinator needs to guarantee that such engineering efforts lead to the intended performance of the integrated system and its user-facing features.

Future research should engage in a more systematic analysis of the actors' information and coordination needs as well as of how to optimize the information flow between them. This is a unique chance for the modeling community, which is called to provide mechanisms for efficient exchange of requirements, design decisions, and models between different organizations. We plan on furthering our discussions with more industrial partners in the automotive industry and broaden as well as validate our understanding of their ecosystem and its challenges, e.g. in the context of Autosar ([www.autosar.org](http://www.autosar.org)). Our long term research goal is propose and develop, in collaboration with industrial partners, solutions towards these challenges.

## Acknowledgements

This work was sponsored by NECSIS and Software Center (an industry-academia partnership, hosted by Dept. of Computer Science and Engineering, Chalmers | University of Gothenburg). We would like to express our special gratitude and thanks to our contacts and interview partners at GM – this work would not have been possible without you!

## References

1. van Angeren, J., Kabbedijk, J., Popp, K.M., Jansen, S.: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, chap. 5: Managing software ecosystems through partnering, 85–102. In: [7] (2012)
2. Bosch, J.: From Software Product Lines to Software Ecosystems. In: Proc. of Int'l Conf. on Softw. Product Lines (2009)
3. Farbey, B., Finkelstein, A.: Software acquisition: a business strategy analysis. In: Proceedings of Requirements Engineering (RE '01). pp. 76–83 (2001)
4. Gao, L.S., Iyer, B.: Analyzing complementarities using software stacks for software industry acquisitions. *Journal of Management Information Systems* 23(2), 119–147 (2006)
5. Jansen, S., Cusumano, M.: Defining software ecosystems: A survey of software platforms and business network governance. In: Int'l WS on SW Ecos. (2012)
6. Jansen, S., Brinkkemper, S., Finkelstein, A.: Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, chap. 2: Business Network Management as a Survival Strategy, pp. 29–42. In: Jansen et al. [7] (2012)
7. Jansen, S., Cusumano, M.A., Brinkkemper, S. (eds.): *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Edward Elgar, Cheltenham, UK (2012)
8. Knauss, E., Damian, D., Knauss, A., Borici, A.: Openness and Requirements: Opportunities and Tradeoffs in Software Ecosystems. In: Proc. of 22nd Int'l Requirements Engineering Conf. (RE '14). Karlskrona, Sweden (2014)
9. Knauss, E., Hammouda, I.: EAM: Ecosystemability Assessment Method. In: Proc. of 22nd Int'l Requirements Engineering Conf. (RE '14). Karlskrona, Sweden (2014)
10. Manikas, K., Hansen, K.M.: Characterizing the danish telemedicine ecosystem: Making sense of actor relationships. In: Proc. of MEDES'13. pp. 211–218. Neumünster Abbey, Luxembourg (2013)
11. Manikas, K., Hansen, K.M.: Reviewing the Health of Software Ecosystems - A Conceptual Framework Proposal. In: Proc. of Int'l Wksp on Softw. Ecosys. pp. 33–44. Potsdam, Germany (2013)
12. Manikas, K., Hansen, K.M.: Software ecosystems: A systematic literature review. *Systems and Software* 86, 1294–1306 (2013)
13. Popp, K.M.: Definition of supplier relationships in software ecosystems as a basis for future research. Tech. rep. (2010), <http://www.drkarlpopp.com/resources/ICSOSubmission2.pdf>
14. Poppendieck, M., Poppendieck, T.: *Lean Software Development*. Addison Wesley (2003)
15. Scacchi, W.: Understanding requirements for open source software. In: Proc. of Design Reqts. Wksp. pp. 467–494. Springer LNBIP 14 (2009)
16. Schultis, K.B., Elsner, C., Lohmann, D.: Architecture Challenges for Internal Software Ecosystems: A Large-Scale Industry Case Study. In: Proc. of 22nd ACM SIGSOFT Int'l Symp. on the Foundations of Software Engineering (FSE '14) (2014)
17. Yantchev, J., Serughetti, M., Lapides, L., Giusto, P.: Session ID #6P17I(Panel) - Intermediate, Simulation: Expert Insights Into Modelling Microcontrollers. Panel, Renesas DevCon (2012), <http://renesasdevcon.com/archives/course.html>, meet the expert, Session ID #6P17I