Publik rapport

Författare: Henrik Lönn
Datum: 2017-08-31
Projekt inom Fordonsutveckling

FFI Fordonsstrategisk Forskning och Innovation

Innehållsförteckning

1 Sammanfattning

Den svenska fordonsindustrin levererar produkter i världsklass. För att upprätthålla sin position måste fordonens funktionalitet, säkerhet och kvalitet vara fortsatt hög, medan ledtid, utvecklingskostnader och produktkostnad kontinuerligt behöver sänkas.
Fordonssystem är mekatroniska, dvs elektronik och programvara interagerar med fysiska komponenter för att leverera funktionalitet. HeavyRoad-projektet har tagit upp metodförbättringar för att säkerställa utvecklingseffektivitet och produktkvalitet.
Projektet bidrar till dessa mål genom att tillämpa virtuell integration och systemsyntes på ett systematiskt sätt, där ingenjörsartefakter delas mellan funktionsutveckling med sitt beteendefokus och systemutveckling med dess komponent- och arkitekturfokus. Genom virtuell integration sätts modeller och kod som motsvarar de faktiska systemkomponenterna samman i ett virtuellt system och analyseras eller simuleras tillsammans med omgivningen. Systemsyntes innebär att tekniska artefakter för det verkliga systemet genereras baserat på befintliga modeller, begränsningar och konfigurationer.

Projektet har fokuserat på att tillhandahålla

- Metodförbättringar avseende kombinerad funktions- och systemutveckling
- Verktyg och koncept för integration av modeller och prototyper av mekatroniska system
- Verktyg och koncept för ökad återanvändning och automatisering under systemutvecklingen

Fördelarna inkluderar

- Ökad fordonsfunktionalitet
  Eftersom nya funktioner kan utvecklas, integreras och valideras snabbt
- Ökad säkerhet och kvalitet
  Eftersom systemintegration och verifiering kan ske tidigt, snabbt och med förtroende
- Minskad ledtid och utvecklingskostnad
  Eftersom automatisering och effektiv metodik minskar resursbehoven
- Minskad produktkostnad
  Eftersom tekniska lösningar snabbt kan utvärderas och optimeras

Projektet har levererat teknologier av följande slag:

- Metodik för utveckling av automationsprogramvara och elektronik inklusive metorder för kontinuerlig integration och representationsmönster
- Verktyg för virtuell integration och simulering av produktionsprogramvara
- Verktyg för virtuell integration och simulering av fysiska komponenter
- Verktyg för prototypprogramvara och elektronik
- Verktyg för representation och specifikation
- Verktyg för syntes och automatisering

2 Executive summary

Automotive systems are cyber-physical, i.e. electronics and software interact with physical components to deliver services. The HeavyRoad project has addressed methodology improvements in order to secure development efficiency and sustained product quality.
This project contributed to these targets by applying virtual integration and system synthesis technologies in a systematic way, where engineering artefacts are shared between function development with its behaviour focus and system development with its component and architecture focus. Through virtual integration, models and code corresponding to actual system components are put together to a virtual system and analysed or simulated together with its environment. System synthesis means that automation is employed to generate engineering artefacts for the real system based on existing models, constraints and configurations.

The project focused on providing

- Methodology improvements regarding combined function and system development
- Tools and concepts for integration of models and prototypes of cyber-physical systems
- Tools and concepts for increased reuse and automation during system development

The project has delivered technologies of the following kinds:

- Methodology for the development of automotive software and electronics including continuous integration approaches and representation patterns
- Tooling for virtual integration and simulation of production software
- Tooling for virtual integration and simulation of physical components
- Tooling for prototyping software and electronics
- Tooling for representation and specification
- Tooling for synthesis and automation

Project technologies has been matured for deployment in production, showing the relevance and need for virtual integration technologies.

3 Bakgrund

The HeavyRoad project has addressed a set of needs currently relevant for Swedish automotive industry. These needs are all related to development methods for electrical and electronic systems, which is a major challenge. To give a perspective, the development of AB Volvo's latest truck architecture was one of the largest industrial efforts in Sweden during the last years. It is of critical importance that such projects have access to the best possible development technology, to control development cost and achieve the best possible product at the right cost. The electrical architecture has a special role, since all functionality is partly or fully based on software. Efficiency improvements of electronics and software development will therefore have a large impact on the product development as a whole.

4 Purpose, research questions and method

The purpose of the HeavyRoad project was to reduce lead time, improve efficiency and enhance product qualities in the context of embedded control systems, and by that the complete product. The HeavyRoad vision has been to align function development with system development. Using continuous integration and delivery, work can be done iteratively and incrementally in a way that involves both functional aspects as well as architecture and component aspects. Virtual integration is a key technology for continuous integration and delivery of mechatronic systems. The research questions of the project are all related to identifying, adapting and deploying technologies in this area.

5 Goals

The purpose of the HeavyRoad project was to increase profitability of Swedish automotive industry by providing technologies that increase development efficiency and increase product quality. The project has defined project objectives related to the themes virtual integration (V1-V3) and system synthesis (S1-S3). These are listed below:

V1. Definition of an approach for modelling behaviour and execution of functional blocks
Enabling composition of simulation models based on connected functional blocks

V2. Plant Model Integration – Vehicle Dynamics and HMI
Integration and co-simulation of systems with plant models for Vehicle Dynamics and HMI

V3. Efficient combined usage of virtual and real targets for verification and validation
Finding appropriate balance and complementarity between V&V activities on virtual and real targets

S1. Generation of AUTOSAR SWC based on functional blocks, their interfaces, behaviour and execution specification
S2. Generation of functional blocks, their interfaces, behaviour and execution specification based on AUTOSAR SWC
S3. Generation of adapters and configurations of rigs based on models of logical and physical architecture

The project also set out three quantitative goals:

a) Reduce the required development effort by 25%
b) Reduce the overall development time by 25%
c) Increase the level of innovation by 20%

The project objectives and quantitative goals were applicable throughout the project and also used to evaluate project results.

6 Results and goal fulfilment

The ability to develop competitive vehicles is increasingly dependent on the appropriate development of the embedded systems. As the authority, criticality and influence of software-based systems increase, vehicle qualities are totally dependent on correct and efficient development of software. The most critical of the software based systems are those interacting with the physical world outside the vehicle. The interaction with mechatronics and multi-physics is therefore a natural part of automotive software development. Below we will elaborate on the HeavyRoad contributions, starting with the workflow and structuring pattern and then continuing with configuration and variability, software simulation aspects and system simulation aspects.

6.1 Workflow

The workflow is part of a continuous integration cycle, with the purpose to automatically verify the (virtually) integrated system in each iteration. There are 4 steps involved:

1. Edit System
   The system is captured as a variable-rich system model where software and models of physical components are integrated. A product instance is defined by selecting among defining features.
2. Resolve Variability
   Based on the feature selection and variant definition, optional components in the variable-rich system model are pruned and parameters are assigned values.
3. Generate Simulation Components
   Behavioral definitions, i. e. code or models depending on component kind, are transformed to simulation modules
4. Simulate and verify
   The simulation components are simulated according to the experiment definitions and verification is performed by checking that stimuli/response is according to stated requirements. The choice of which product to instantiate and which requirements to verify is part of the (automated) experiment planning and not further discussed in this paper.

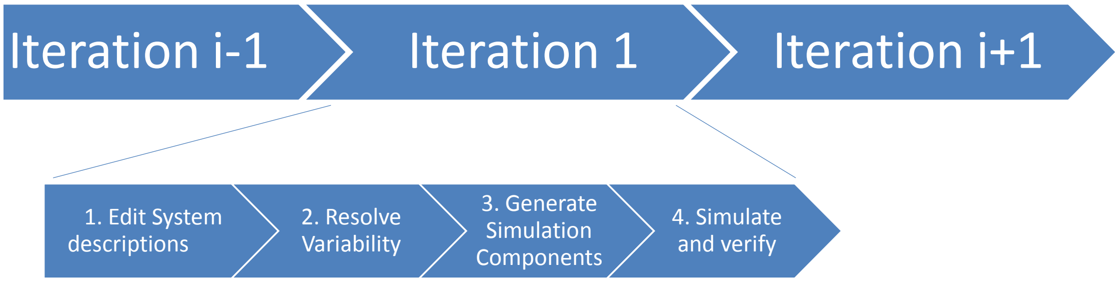

Fig. 1. Overall Workflow.

6.2 Representation Pattern

The chosen pattern for the information model is based on the SimArch simulation architecture [17] and extended based on EAST-ADL [6] concepts. We have contributed with a refined representation of structure and behavior and added concepts for plant interfacing, see Fig. 2. The pattern distinguishes between five parts: i) an application part that is represented by (production) software or its corresponding models, ii) an I/O part representing sensors, actuators and electrical interfacing, iii) a plant that represents the in-vehicle physical elements, iv) an environment that represents elements outside the ego-vehicle and finally v) the stimuli and expected response for the purpose of specification and verification.

By applying this pattern and maintaining the corresponding interfaces, it is possible to reuse the model parts for different verification use cases. The unidirectional arrows in Fig. 2 show suitable interfaces for such experiments. The leftmost arrow (1) represents the boundary of application software corresponding to an engineering units interface for application software-in the loop and model-in-the loop. The boundary may be extended to the electrical interface (2) and thus include the sensor/actuator abstraction. The software boundary (3) includes the platform and is aligned with processor-in-the-loop or virtual targets for target-compiled binaries. The control unit boundary (4) is the electrical interface to sensors and actuators and corresponds to hardware-in-the-loop. The sensor or actuator boundary (5) is the physical interface between the complete embedded system and corresponds to e.g. rapid control prototyping, i.e. where preliminary control systems are used in a mule truck. Finally the vehicle boundary (6) is the external interface of the vehicle, i.e. "tires-to-road".

Examples of how modeling elements are reused in different use cases include running a real control unit together with models of the sensors, plant and environment, or executing binaries on a virtual processor together with models of electronics, and the same models for sensors/actuators, plant and environment. Similarly, models of the embedded system can be connected to sensors and actuators in a real truck and tested in the field.

It should be noted that the granularity can vary depending on the purpose of the experiment. For example, sometimes the I/O part is simplified and not decomposed into sensor/electronics/sensor software while in other cases also the electronics part may be decomposed into wiring harness and discrete electronic components.

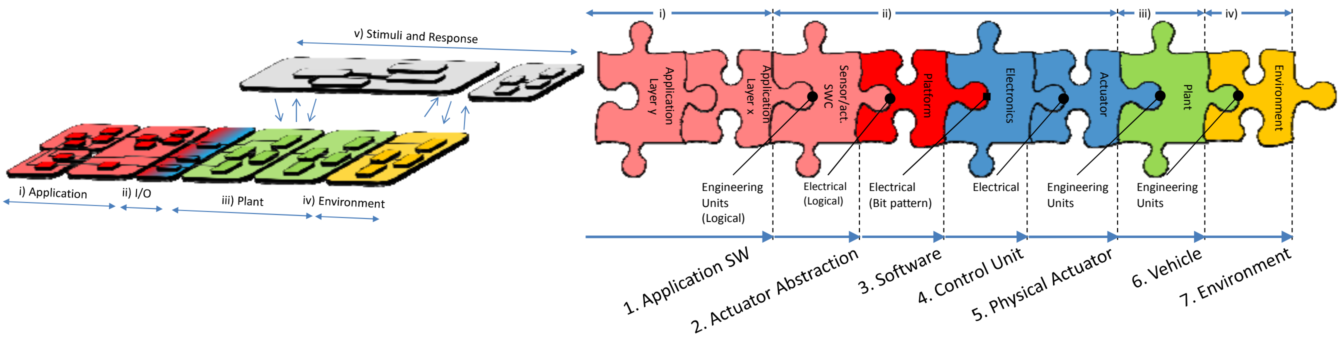

Fig. 2. Modelling Pattern for Embedded System, with interfaces at appropriate locations

6.3 Variability and Configuration

In the overall workflow, as discussed in Section **Error! Reference source not found.**, variability and configuration is a key aspect. In order to promote reuse and secure separation of concern, variability is defined in three dimensions:

1. Product: The feature content of the product influences how the software and embedded system are configured, as well as the models of physical components.
2. Experiment: Depending on which requirements to verify and which experiment setup is deemed appropriate for a given verification effort, different simulation components need to be selected and parameterized accordingly.
3. Notation/Representation: Different behavioural representations are suitable depending on the intended simulation target and fidelity.

The variant selection in each of the dimensions is done in order to support the verification effort at hand.

6.4 Simulator

Virtual integration is often performed with the purpose to simulate a representation of the system. Behavioral modeling tools typically provide simulation capabilities of components, but there is limited support for system aspects such as execution and communication coordination beyond component level. In such simulations, concurrency, contention and interaction among components are typically not represented. To include these aspects, architecture models provide system descriptions and a general execution framework provides for simulations that respect timing, triggering and concurrency aspects.

There are several such off-the-shelf integration frameworks for the automotive domain such as Silver [13], CANoE [14] and Scalexio [5]. However, such tools are mainly intended for interactive, desktop use rather than large scale engineering automation. Further, they are difficult to tailor for company specific needs or refine towards new capabilities concerning work flow, representation, test execution, etc. These issues were investigated in the HeavyRoad project, resulting in a flexible and efficient simulation environment.

A specific simulation platform was developed in the HeavyRoad project, with specific attention to the Volvo context and the complex experimental setting and continuous need for tailoring. Fig. 3 shows the architecture of the Adapt simulation platform. The simulation core is responsible for triggering simulation modules and data exchange over a simulation bus. Adapt modules represent software components, physical simulation models, logging, and even interfaces to physical buses and I/O.

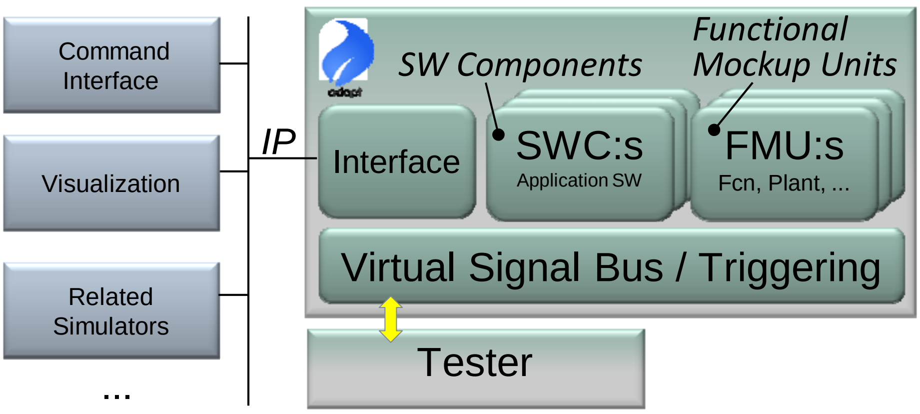

Fig. 3. ADAPT Simulation platform

6.5 Fidelity of Plant and Environment

In order to integrate simulation models of physical phenomena, the Functional Mockup Interface is used. When accuracy requirements are low, it is possible to run the Functional Mockup Units as individual simulation modules, only executing and exchanging data on specified points in time according to its trigger conditions. There is also the possibility to integrate several Functional Mockup Units to an aggregate that internally is simulated with a high precision solver. This way, tightly coupled components can be simulated with preserved precision and accuracy, even if the external simulation is running at slower speed. The definition of the aggregate is done using the System Structure and Parameterization format, and a solver synthesis approach is defined in the project.

6.6 Fidelity of Embedded System Simulation

The embedded system can be simulated based on models of software, electronics and sensors and actuators. They are then represented as Functional Mockup Units. The physical elements largely have the same implications as discussed above. When AUTOSAR software components are executed on the Adapt simulator, modules are generated based on the original application code and autogenerated wrapper code. The wrapper code is a minimal replacement of the AUTOSAR middleware to execute the application. To increase fidelity of the software execution and to include the platform configuration it is also possible to run AUTOSAR application code on a Virtual AUTOSAR Cluster, VAC. The application code is then invoked exactly as in the product, and the full middleware is executing. Except that code is cross-compiled and the drivers for I/O and communication is modified, full execution fidelity is achieved, and the tooling is fully consistent with the target tooling.

6.7 Validation and Verification options

An integrated system simulation model can be used for both automatic black box accelerated testing and for real-time interactive validation. The former is perhaps the most important capability, in order to verify dangerous and rare driver situations and collect enough testing time to reach required confidence levels. In this simulation mode, test stimuli and response analysis, as well as invariant monitoring is performed without user interaction. The Adapt platform also supports real time execution. In this simulation mode, visualizations can be used to interact with operators, test subjects and other stakeholders, see **Fig. 4**. Integration with physical I/O and buses is also possible, using the corresponding interface modules to support rapid control prototyping or hardware-in-the-loop.

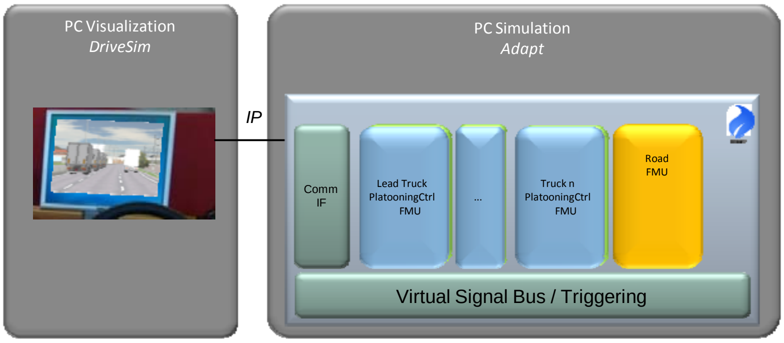

Fig. 4. Adapt platooning simulation together with visualization tools

6.8 Tooling

Various tooling has been developed to support the different parts of the workflow identified in the project. SystemWeaver is an enterprise database product that has been adapted to support product models according to Section 6.2. Similarly, the EATOP/Artop open source platform denoted ArEATOP has been refined during the project. Both tool environments have been expanded to support modeling, variability definition and resolution and simulation generation.

To support the fidelity concepts in Section 6.5, the Modelon Functional Mockup Unit Aggregation tooling [11] was developed. Similarly, the Virtual AUTOSAR Cluster concept in Section 6.6 was made as an extension to Arccore tooling for AUTOSAR [1].

6.9 Example System

In order to validate project results, several example systems have been modeled and simulated. One of them is a platooning system, where the truck platoon was simulated on the Adapt platform and positions were provided to a visualization tool, to allow interactive validation. Prior to simulation, the variable rich system models were resolved for variability and simulation modules generated from the component descriptions.

6.10 Project Results vs. Programme and Project Goals

Simulation based verification is an indispensible technology for developing embedded automotive system. It is needed to support continuous integration and deployment with sustained confidence, and it is essential in verifying correct behavior in rare and dangerous situations. HeavyRoad has addressed various aspects of integration and simulation, and thereby contributed to programme goals and delivered on the project goals.

Contribution to Overarching FFI objectives

- Increasing the Swedish capacity for research and innovation, thereby ensuring competitiveness and jobs in the field of vehicle industry
  Improved development methods for embedded systems is a key enabler for increasing the pace of innovation. Industrialization of new ideas driven by technological opportunities and user demand are currently hampered by engineering capability. Improved methods for electronics and software development resulting from HeavyRoad is a way to relieve this constraint.
- Developing internationally interconnected and competitive research and innovation environments in Sweden
  The HeavyRoad project has leveraged on open standards such as EAST-ADL, AUTOSAR and FMI. Existing research networks has been involved in the activities, and the focus on non-proprietary solutions has allowed external and international collaborations.
- Promoting the participation of small and medium-sized companies, SME
  Three of the HeavyRoad partners are SMEs and have been key contributors in addressing the FFI objectives. The use of open standards in the results is an anabler for further participation of SMEs.
- Promoting the participation of subcontractors
  The virtual integration concept developed in the project is based on open standards and allows a heterogeneous set of components to be integrated. The technology is therefore highly suitable for efficient supplier-manufacturer interaction.
- Promoting cross-industrial cooperation
  While parts of the project results are based on automotive standards, most notably those related to software and architecture modeling and software platforms, the virtual integration concepts are largely domain independent and applicable in other industries. Being related to software development, continuous integration and deployment is cross-industrial in nature.
- Promoting cooperation between industry, universities and higher education institutions
  The project has promoted and participated in academic activities related to modeling and simulation. The collaboration is sustained by the use of standards and open source platforms and components.

Contribution to sub-programme objectives

The specific objectives for Vehicle Development – Enabling Electronics are also addressed within HeavyRoad. An “Increased ability by participants to understand and manage development of future platforms, which increasingly are based on electronics and software” depends heavily on processes, methods and tools addressed by this project. Similarly, the goals to enhance

- Technology and method development for reduced environmental impact, reduced energy consumption, increased traffic safety and increased competitiveness
- Capacity to develop complex electrical systems

- Capacity to develop complex embedded systems

were all addressed by HeavyRoad. The first goal is implicitly addressed through the improved capability regarding virtual integration and simulation of new functionality – it is possible to develop better products if development iterations are quicker and if verification and validation can be done also on virtual systems. HeavyRoad results regarding simulation of physical components like sensors and actuators together with software enhance the capacity to develop both complex electrical systems and complex embedded systems.

Project objectives – Virtual Integration

V1. Definition of an approach for modelling behaviour and execution of functional blocks
Enabling composition of simulation models based on connected functional blocks

√ An architecture description for software and system elements has been used, based on the EAST-ADL and AUTOSAR languages. Behavioural definitions were captured using the FMI standard for functional components and AUTOSAR-aligned source code for software components.

V2. Plant Model Integration – Vehicle Dynamics and HMI
Integration and co-simulation of systems with plant models for Vehicle Dynamics and HMI

√ Integration of controllers and vehicle dynamics plant is captured using EAST-ADL, AUTOSAR and FMI as described for V1. The integrated system is simulated using the Adapt virtual integration platform developed in the project. HMI panels and controls were defined in the Qt framework. With gateway functionality, the software and system simulation could be integrated with Qt panels.

V3. Efficient combined usage of virtual and real targets for verification and validation
Finding appropriate balance and complementarity between V&V activities on virtual and real targets.

√ The test interface of the virtual test environment developed in HeavyRoad is equivalent to the test interface of physical rigs. In cases where physical rigs relied on simulation models, the same approach was used for the virtual rig. For this reason tests and results were fully portable across the environments.

Project objectives - System Synthesis

S1. Generation of AUTOSAR SWC based on functional blocks, their interfaces, behaviour and execution specification

√ Model transformation from EAST-ADL functional blocks to AUTOSAR software components have been implemented.

S2. Generation of functional blocks, their interfaces, behaviour and execution specification based on AUTOSAR SWC

√ Model transformation from AUTOSAR software components to EAST-ADL functional blocks have been implemented.

S3. Generation of adapters and configurations of rigs based on models of logical and physical architecture

√ Interface hardware and corresponding modelling and generation support have been integrated and developed for CAN, LIN and TCP/UDP

Measurable Goals

The measurable goals of HeavyRoad are related to the qualitative goals listed above. Below we will comment on how each of these are met. Use of HeavyRoad technology will

a) Reduce the required development effort by 25%
HeavyRoad technology has automated several engineering steps which clearly contributes to a reduced development effort. In addition, because the loop time for changes could be reduced there is also a secondary effect that is significant: When corrections are needed, these can be deployed before other, possibly conflicting, system changes has been introduced in the product, thus eliminating double work. Increased automation together with reduced rework allows reduction of the development effort of more than 25%.

b) Reduce the overall development time by 25%
HeavyRoad technology has automated several engineering steps and allow an increasing part of integration and verification to be done on virtual vehicles, cutting time on prototyping and testing in rigs and real vehicles. This component alone accounts for a time reduction of more than 25%. Adding to this

comes time gained on shorter correction loops when design flaws are identified or when the intended scope and functionality change.

c) Increase the level of innovation by 20%
Virtual integration and simulation provides fast feedback on inventions, allowing rapid improvements and enhancements. It also allows an increased understanding of the vehicle functions, thus promoting generation of new ideas and improved functionality.
Clinics has been performed with engineers using HeavyRoad technology, demonstrating that new functionality could be conceived, detailed and implemented in a significantly shorter time than today. Extrapolating on these experiences, the number of innovations can increase well beyond 20%. Business, production and maintenance aspects will also influence the permitted innovation rate, but from a technical perspective project results exceeds this goal.

7 Dissemination

HeavyRoad results have been presented, demonstrated and discussed in several events throughout the project time. Fig. 5 shows pictures from the joint final event between FFI projects HeavyRoad and SecondRoad.

Fig. 5. Pictures from final Event

7.1 Knowledge and Results Dissemination

How has/will the project outcome be used and distributed?		Comment
Increased knowledge in the field	X	The project has resulted in knowledge about methodology and the specific possibilities and limitations with standards and methods in virtual integration and simulation. Within the companies, engineers have been involved in the work with HeavyRoad, thus developing skills in the area of the project.
Transfer to other advanced engineering projects	X	Project results are used as the basis for further refinement in partner internal projects, ECSEL projects and FFI projects such as Open Innovation Lab and SimulationScenarios.
Transfer to product development projects	X	Project results are used as part of regular product development in integration and verification phase
Market Introduction	X	Project results are part of/planned to be part of

		commercial products from the project's tool suppliers. Technologies are provided as knowledge as part of the partners' service offers.
Used in investigations / regulations / concessions/ political decisions		

HeavyRoad has developed prototype tools for modeling and synthesis of simulations, that will be used in the FFI projects SimulationScenarios and Open Innovation Lab. The project proposal ECSEL HybridMDE represents another (potential) recipient of the project results.
Internally, project partners continue work to refine and industrialize project results. Several activities are underway aimed at streamlining integration and testing of software-based systems and rationalizing simulation-based evaluation.

7.2 Publications

Table 1. Publications and presentations

Presenter/Authors	Title/Topic	Venue	Location	Date	Kind
Henrik Lönn	FMU Technology	Second Road FMU Technology Day, March 2015	Torslanda	150301	Pres
Peter Thorngren	Continuous Integration as a step in the Stairway to Heaven Vision	Continuous Delivery & DevOps Conference, November 2015	Stockholm	150601	Pres
Peter Thorngren	Continuous Integration. Why software/hardware integration matters in commercial vehicles now and why it will be critical in the future – techniques and methods	Vector UK Conference, June 2015	Birmingham	150601	Pres
All Partners	HeavyRoad Overview	SecondRoad Bazaar, September 2015	Gothenburg	150826	Pres
Volvo	HeavyRoad Overview	Volvo GTT/ATR Tech Townhall	Gothenburg	150902	Pres
Peter Thorngren	TBD	Continuous Delivery & DevOps Conference, November 2015	Copenhagen	151101	Pres
Henrik Lönn	Variants vs. Embedded System Testing, Simulation and Analysis	Smart Automotive Variantcon	Berlin	151111	Pres
Volvo	HeavyRoad Overview	Volvo Group Tech Expo	Gothenburg	151125	Pres
Volvo	HeavyRoad Overview	Volvo GTT/ATR/EES Tech Day	Gothenburg	151208	Pres
All Partners	HeavyRoad Demo	SecondRoad Bazaar, January 2016	Torslanda	160122	Pres
All Partners	HeavyRoad Demo	VICTA Innovation Bazaar	Gothenburg	160204	Pres
Peter Thorngren	Enabling Creativity – the truck system with Continuous Integration and Virtualization demonstrated.	International Software Development Conferenc	London	160301	Pres
Peter Thorngren	Continuous Integration	VICTA Roundtable	Gothenburg	160406	Pres
Henrik Lönn, Henrik Kaijser &	HeavyRoad Overview	Elektronik i Fordon	Gothenburg	160511	Pres

Saimir Baci					
Martin Ivarsson	Variability mgmnt and PLE for embedded systems	Elektronik i Fordon	Gothenburg	160511	Pres
Henrik Lönn	HeavyRoad Overview and Demo	Synligare/FUSE Seminar May 18th	Gothenburg	160518	Pres
Maria Henningsson et. al	The Path Towards Virtual Development and Verification Using FMI	JSAE, Japan	Japan	160525	Paper
Peter Thorngren	Supporting creativity new business solutions with improved testing and integration capabilities	Annual Automotive Embedded Multi-Core Systems Summit	Stuttgart	160514	Pres
Peter Thorngren	Integration and Test	Internet of Things, Oslo	Oslo		Pres
Kaijser, Lönn, Thorngren	Virtual Integration on the Basis of a Structured System Modelling Approach	EDCC/CARS 2016	Gothenburg	160906	Paper
Blom et. al	EAST-ADL – An Architecture Description Language for Automotive Software-intensive Systems in the Light of Recent use and Research	IGI International Journal on International Journal of System Dynamics Applications	N/A	160701	Paper
Henrik Lönn	Heavyroad Demo - VCC vOBC@Adapt	SecondRoad Bazaar, August 2016	Gothenburg	160830	Pres
Peter Thorngren	Continuous Integration	We-conect Efficient App Development	Berlin	160920	Pres
Volvo	Trucks and HeavyRoad demos	Lindholmen SW Development Day	Gothenburg	161026	Demo
Henrik Lönn	Variant management for a Virtual Integration framework	Smart Automotive Variantcon	Berlin	161122	Pres
Volvo	HeavyRoad Overview and Demo	GTT Days	Gothenburg	170516 -17	
Project Partners	HeavyRoad Overview and Demo	HeavyRoad Event	Gothenburg	170530	Pres
Kaijser, Lönn, Thorngren, Ekberg, Henningsson, Larsson	Towards Simulation-Based Verification for Continuous Integration and Delivery	ERTS Submission	N/A	In Submission	Paper
Henrik Lönn	HeavyRoad FMI Usage	ITEA FMI Success Stories	N/A	To appear	Article

8 Conclusions and further research

The HeavyRoad project has delivered a set of conceptual and concrete results. Below, we will comment on some of these regarding status and opportunities of further refinement.

- Adapt Simulation Platform
  The Adapt platform manages execution and data transfer between simulation agents. Further refinements may include different management of events, support for asynchronous and event-based execution.
- Adapt AUTOSAR SWC module
  The AUTOSAR module generator is currently supporting relevant AUTOSAR 3 and 4 constructs. As the use of AUTOSAR evolves over time, continued refinement is required.
- Adapt FMU execution module

The FMU execution module converts FMU:s according to the FMI standard to Adapt modules. Further refinement of the FMU module involves

- Adapt CAN/LIN/IP/IO module
  This module provides external interfacing to Adapt based on different I/O and bus technologies. The module is a prototype that will require further refinement and validation to be reliably used in production.
- Adapt StructuredText module
  The StructuredText module generator provides Adapt modules based on PLC structured text definitions. The generator is well proven and no specific evolution needs are foreseen.
- Adapt custom modules
  Several custom modules respecting the Adapt API are available, and more modules will be added as simulation needs occur.
- Adapt Studio debugging environment
  Signal inspection and simulation interaction is provided by the debugging environment. Continuous refinements will be made to support user needs.
- Adapt/CITA test automation
  The CITA test automation module runs test scripts and evaluates results in the Adapt environment. As test methods are refined over time, and the test targets change, the CITA module will need to evolve along with it.
- Adapt Graph Creator
  Simulation signals are visualized on diagrams. The user interface and diagramming capabilities are prototypical and further enhancements are possible.
- Arccore Virtual Autosar Cluster
  Virtual Autosar Cluster allows simulation of AUTOSAR 4.0 components on Arctic Core, i.e. the actual embedded platform that is extended with interface modules towards the Adapt API. Further evolution of the platform entails adding support for more I/O and comm interfaces towards Adapt.
- Arccore ReDapt logging and replay
  The ReDapt module allows recording of Adapt virtual simulation bus signals and replay of PCAP log files onto the virtual simulation bus. Currently, the number of signals allowed is limited and further work is needed to extend this.
- ArEATOP Adapt Builder
  Adapt builder generates simulation artefacts for Adapt from AUTOSAR, EAST-ADL and FMU based architecture models. The tool is experimental and further refinements regarding usability and functionality is foreseen. Because the tool is open source and Eclipse based, it is amenable for collaborative research.
- ArEATOP AUTOSAR – EAST-ADL converter
  The converter plugin converts AUTOSAR software components to EAST-ADL functions and vice versa. There are refinement opportunities regarding the mapping strategy of the current plugin.
- ArEATOP FMU Import
  The FMU import plugin on ArEATOP, generates an EAST-ADL function for the imported FMU, and links the function behavior to the FMU. Certain refinements in the user interface and introduction of alternative mapping rules can be expected.
- ArEATOP SSP Export
  The SSP Export plugin for ArEATOP generates an FMI SSP file based on an EAST-ADL composite function.
- ArEATOP Variability Resolution
  The variability plugin allows variability based on EAST-ADL variability concepts to be defined and resolved. Further refinement can be foreseen in visualizing variants and supporting the variability definitions.
- Modelon FMIx FMU Aggregation tool

FMIx is a conversion tool that creates a composite FMU with an integral solver based on a set of connected FMUs. Further refinements and options for the solver are planned as the tool is commercialized.

- Modelon FMI Composer
  FMI Composer is a graphical editor for the connected FMUs, allowing FMU aggregates to be defined and inspected before aggregates are generated.
- Systemite System Weaver Adapt plugins
  SystemWeaver support for defining and resolving variant-rich controllers and plants,and generation of Adapt artefacts.The modeling and generation can be further refined in the context of larger system examples and full scale engineering workflow.
- Qt based visualization
  Qt based visualization was prototyped using CAD images and 3d-geometries. Further work is required to refine the solution and allow more examples to be visualized. Alternative rendering hardware is another refinement option.

In addition to the concrete tools and technologies listed above, the project has worked with methodology, modeling patterns and continuous integration environments to sustain efficient software and system development. Overall, the project has contributed with key technologies that are critical for sustaining the Volvo products. A consortium has been formed with the purpose to share Adapt technologies and share the effort of further evolution of the simulation environment. The Founding members are AB Volvo and Volvo Cars Corporation AB, and further members will be invited over time.

9 Participating Partners and Contact Persons

	Volvo Technology AB Henrik Lönn
	ArcCore AB Johan Ekberg
	Modelon AB Maria Henningsson
	Systemite AB Mats Larsson

10 References

[1] Arccore: AUTOSAR platform and tooling. http://www.arccore.se

[2] Architecture Analysis & Design Language, SAE Standard AS-5506

[3] Autosar Development Partnership: AUTOSAR 4.2, www.Autosar.org

[4] AUTOSAR Tool Platform User Group: Artop. http://www.artop.org

[5] DSpace Scalexio – product information. http://www.dspace.com

[6] EAST-ADL Association: EAST-ADL 2.1.12, www.east-adl.eu

[7] Eclipse: EAST-ADL Tool Platform. www.eclipse.org/eatop

[8] FMI development group: FMI 2.0, www.fmi-standard.org

[9] H. Kaijser, H. Lönn and P. Thongren: Virtual Integration on the Basis of a Structured System Modelling Approach. EDCC CARS, Sweden, 2016

[10] IEEE: Standard 42010, Systems and software engineering — Architecture description.

[11] Modelon AB: FMI Composer tooling. http://www.modelon.se

[12] Object Modelling Group: OMG Systems Modeling Language, www.omgsysml.org

[13] QTronic Silver – Product Information. http://www.qtronic.com

[14] R. Gao, M. Henningson, J. Andreasson and H. Kaijser: The Path Towards Virtual Development and Verification using FMI. JSAE, Japan 2016.

[15] SystemWeaver – Product Information. http://www.systemweaver.com

[16] Vector CANoe – product information. http://www.vector.com

[17] Vinter, Jonny: Simarch Final Report, V-ICT Project SimArch, Sweden, 2010

[18] HeavyRoad consortium: Deliverable D1.1, 2016

[19] HeavyRoad consortium: Deliverable D2.1, 2017

[20] HeavyRoad consortium: Deliverable D2.2, 2017

[21] HeavyRoad consortium: Deliverable D2.3, 2017

[22] HeavyRoad consortium: Deliverable D3.1, 2017

[23] HeavyRoad consortium: Deliverable D3.2, 2017

[24] HeavyRoad consortium: Deliverable D4.1, 2017

[25] HeavyRoad consortium: Deliverable D4.2, 2017