Proceedings of the 2011 IEEE
International Conference on Mechatronics and Automation
August 7 - 10, Beijing, China

Model Driven System Engineering for Vehicle System utilizing Model Driven Architecture approach and Hardware-in-the-Loop Simulation

Seng Chong, Chi-Biu Wong, Haibo Jia, Hongtao Pan and Philip Moore

Mechatronics Research Centre
Department of Engineering
De Montfort University
Leicester, UK

{skchong, cbwong, hjia, hpan & prmoore}@dmu.ac.uk

Roy Kalawsky and John O'Brien

Research School of Systems Engineering
Systems Engineering Innovation Centre
Loughborough University
Leicestershire, UK

{r.s.kalawsky & j.t.obrien}@lboro.ac.uk

***Abstract* – In this paper, we show how Model Driven System Engineering (MDSE) can be applied in the development of a vehicle system. The drivers for the research have been to increase efficiency, flexibility and reconfigurability of UK vehicle system engineering, for enhanced competitiveness and innovation in face of increasing global competitions. Key contributions of the research include (i) establishment of baseline Model Driven Architecture (MDA) approach at the platform independent level; (ii) identification of suitable methodology and toolset (based on Rhapsody / Simulink combination) to permit MDA system designs and hardware-in-the loop (HIL) simulation (based on xPC Target by Mathworks) for system verification; (iii) design and development of a model transformer for seamless transformation of the Rhapsody model to a Simulink model. The end outcome is a streamlined, integrated development platform to facilitate integration of various subsystems and verification of overall system performance to meet the desired system requirements. A case study based on a vehicle system (in this case a train system) had been implemented to verify the effectiveness and capability of the development platform by designing and developing an engine controller and an ABS controller for the train system. The research outcomes shown promising results, which are anticipated to benefit the wider transport sectors, as well as other sectors such as manufacturing and automation.**

Index Terms* – *Model driven system engineering (MDSE), Model driven architecture (MDA), hardware in the loop (HIL) simulation, and model based design.

I. Introduction

Current vehicle architectures and systems are generally tied to specific technologies which make them less adaptable for future changes or upgrades and inefficient in terms of application and manufacturing re-use.

The research has addressed how model driven architectures when coupled with open systems architectures can be used to support future vehicle designs by driving the system architecture from a platform independent perspective. A key aspect is the generation of the system architecture and design that is both platform independent and which can be migrated to a platform dependant solution.

The adoption of open standards, rather than closed proprietary architectures, is an essential economic driver for opening closed markets to competition and innovation.

The proposed research program was exceptionally timely and novel in that it has addressed the use of model driven architectures for future vehicular systems. The rail sector is a particularly appropriate first adopter for the research output because the train system is inherently complex. In addition, trains generally have a longer lifespan and may require technology updates during their useful operating lifespan – this makes cost effective technology refreshes highly desirable.

A key feature of the proposed research has been to demonstrate an end-to-end mapping of a MDA with a demonstration of the benefits. This will push the boundaries and increase our capability of MDA based systems engineering using novel designs, considering design optimization, system performance, and process performance in order to arrive at the most efficient and cost effective solution.

The work was intended to contribute towards the efforts of industry to produce the 'next generation of advanced vehicle technologies'. It was carried in a three phases as illustrated in Fig. 1.

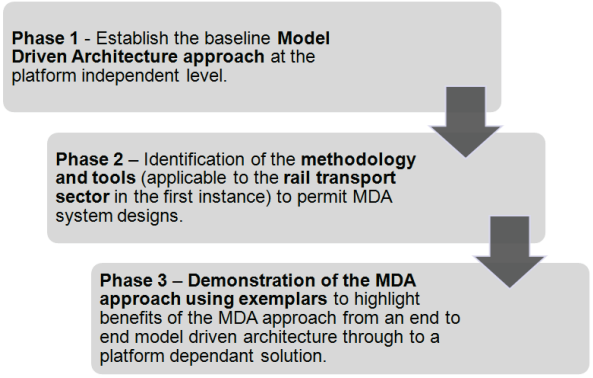

Fig. 1 Three phases of the iNET project.

The project was led and managed by the Research School of Systems Engineering, Loughborough University. They were also responsible for development of the platform independent SysML Models (engine controller and ABS controller) using the IBM Rational Rhapsody software.

The Mechatronics Research Centre, De Montfort University was the academic partner responsible for the model transformation from Rhapsody to Simulink, and hardware-in-the-loop (HIL) simulation based on xPC platform from Mathworks Inc. The project period was six months in total (October 2009 - April 2010), there were a relatively larger number of researchers involved due to the complexity of work and tight schedule.

II. MODEL BASED SYSTEM ENGINEERING

A. Model Driven Architecture (MDA)

Model-based systems engineering has been adopted by the Object Management Group (OMG) as Model Driven Architecture (MDA). MDA is a contemporary software development approach aiming to shift conventional code centric software development to model-centric software development. The development process involves specifying system functionality with a platform-independent model (PIM) which can be converted into platform-specific model(s) (PSM) that allows fully executable source code to be generated as working implementation [1, 2, 3].

The three primary goals of MDA are portability interoperability and reusability through architectural separation of concerns [3, 4]. Extending these platform independent models through the MDA process provides early evaluation of the system and verification of system compatibilities reducing overall development cost [5, 6]. Literature search shown that UML and SysML are indispensible tools for MDA development [7, 8, 9, 10].

B. Methodology and Tools for MDA Approach

The IBM Rhapsody development environment (based on UML and SysML) is one of the most comprehensive MDA methodology and tools currently available. To support the MDA development process, IBM has created the Rational Harmony (Fig. 2) as guidelines to provide a workflow and a library of best practices for building better systems and software.

In addition, Rhapsody has created five specific diagrams (Systems, Software Component, Internal Behavior, ECU and Topology) to support the specific requirements of the AUTomotive Open System Architecture (AUTOSAR), which is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers [5].

The Rhapsody / Simulink combination is powerful because it offers a complete process coverage solution while allowing users to select the tool most appropriate for each activity in the design cycle.

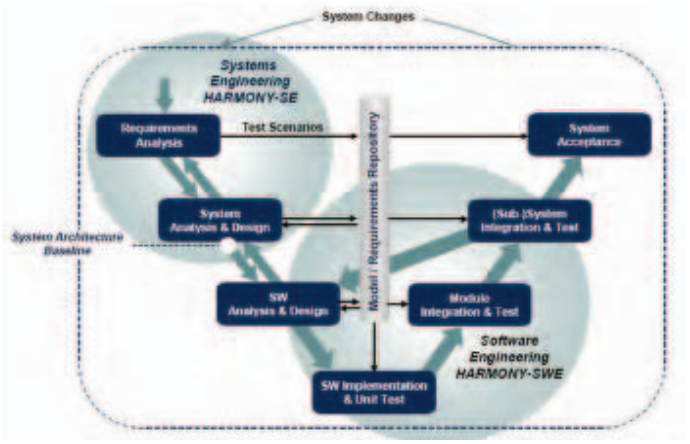

Fig. 2 The IBM Rational Harmony

With Rhapsody, users can access requirements, capture and analysis functions, define systems and software architecture and develop logical algorithms. Each function can be simulated and tested to ensure that they are correct; customized documentation can be generated for enhanced communication and the actual implementation can be realized through production code generation. It is worth noting that Rhapsody can also rapidly target the platform independent application model to a real time embedded operating system (RTOS). It uses XMI as neutral format for interchange of objects and models with third party systems.

With Simulink, users can create system models that define the dynamic behavior of the physical elements that the system will interact with and hence develop the mathematical algorithms that control these system models. This can all be simulated in Simulink to ensure proper results; code can be generated from the control algorithms for use in controller prototyping and production, and from the system models to be used in hardware-in-the-loop (HIL) testing [11].

C. Hardware-in-the-Loop (HIL) Simulation

Hardware-in-the-Loop (HIL) Simulation has been used widely in a range of fields, particularly in the aerospace and automotive industry [12]. However, this technique is by no means limited to those industries, as they can be very useful wherever practical tests of system components are necessary prior to final construction and deployment.

As Rhapsody and Simulink combination was identified as the development platform for the research, it was a natural choice to employ the Simulink xPC Target for execution of Simulink models on a target computer for rapid control prototyping, hardware-in-the-loop (HIL) simulation, and other real-time testing applications. It also provided a library of much needed drivers, a real-time kernel, and a host-target interface for real-time monitoring, parameter tuning, and data logging.

The literature review and finings from initial investigation helped confirm and inform further research work on two of the main project challenges, i.e. (i) identification and integration of an existing toolset to function as a coherent platform to support the model driven architecture; and (ii) verification and demonstration of the integrated platform from a selected case study using hardware in the loop simulation technique.

III. Model Transformation

At present, Rhapsody and Simulink complement each other well and allow bi-directional integration in the following manner [13]:

- Simulink algorithms integrated as a block within Rhapsody model
- Rhapsody design as a block within Simulink Model

While Rhapsody and Simulink are able to exchange models bi-directional by importing into each other's model as a function block (co-simulation) [10, 14]. The main limitation remains the capability to transform a Rhapsody model seamlessly to a Simulink model [5, 11, 13]. This 'model transformation' task was identified as major part of the project work; because a transformation tool did not exist, which was crucial for further exploitation of the collective advantages of the Rhapsody / Simulink combination.

The guidelines from Rhapsody [15] revealed that SysML leverages the OMG XML Metadata Interchange (XMI) to exchange modeling data between tools. In fact, Rhapsody provides an easy way to export their SysML models to an XMI format which is an XML based text file. Further research helped identify the use of a template based model transformation approach for model transformation. The model transformation process devised from the research can be simplified as shown in Fig. 3.

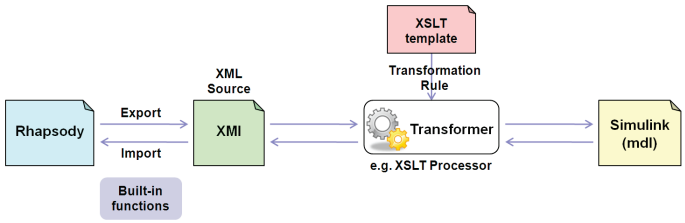

Fig. 3 Model Transformation based on XMI

In this case, the XSLT (Extensible Stylesheet Language Transformations) was used as transformation template, to define the meta-model level transformation between these two models. The XMI file generated from Rhapsody and the XSLT transformation template defined are then input into the XSLT execution engine which was developed as part of the project, to execute the meta-model level transformation of a SysML model to a Simulink model (mdl file).

Within Rhapsody, the ***block definition diagram*** was used to describe the train system hierarchy and system/component classifications. This internal block diagram helped describe the internal structure of the train system in terms of its parts, ports, and connectors. To model behavior of the train system, ***statecharts*** were used for defining the behavior of objects, i.e. the various states of an object and the messages or events that cause it to transition from one state to another.

In fact, the model transformation carried out in this project had been focusing on converting block definition diagrams and statecharts from Rhapsody to Simulink block diagrams and stateflow charts. The Simulink ***stateflow*** chart was used as it provides (i) the language elements for describing the complex logic in a natural, readable, and understandable form, and (ii) an efficient environment for designing embedded systems that contain control, supervisory, and mode logic.

IV. Case Study – Vehicle System

In order to demonstrate and verify the capability of the MDA platform identified and integrated, a case study was employed by creating an engine controller and an anti-lock braking system (ABS) controller for a vehicle system. The engine controller is running on timing and controlled by the 'desired rpm' parameter via user input in the form of throttle (an analogue input). The controller will then alter the throttle angle of the vehicle depending on desired rpm, while monitoring the cranks speed of the engine at the same time. On the other hand, the ABS used a 'bang-bang' controller to maintain a stable wheel's normalized relative slip by monitoring the wheel speed and vehicle speed under hard braking conditions in order to control the braking actions. The development process can be summarized as in Fig. 4, which involved developing PIMs (engine and ABS controllers) in Rhapsody from the system requirements. The PIMs were then transformed into platform independent Simulink models using the in-house developed transformer. These models were then customized based on the hardware platforms on which they will execute, in this case the PIC24 microcontroller and LabVIEW CompactRIO hardware platform.

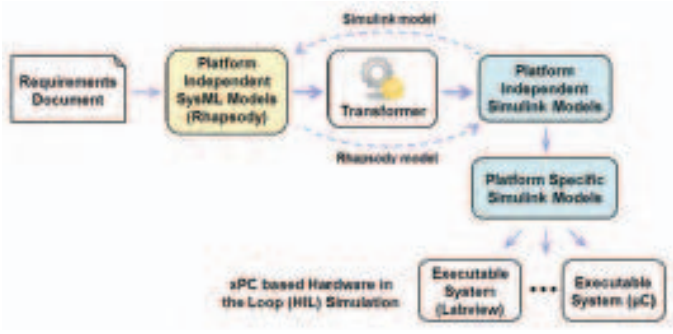

Fig. 4 The development process

A. Platform Independent Model

Fig.5 show the platform independent engine controller model developed in Rhapsody.

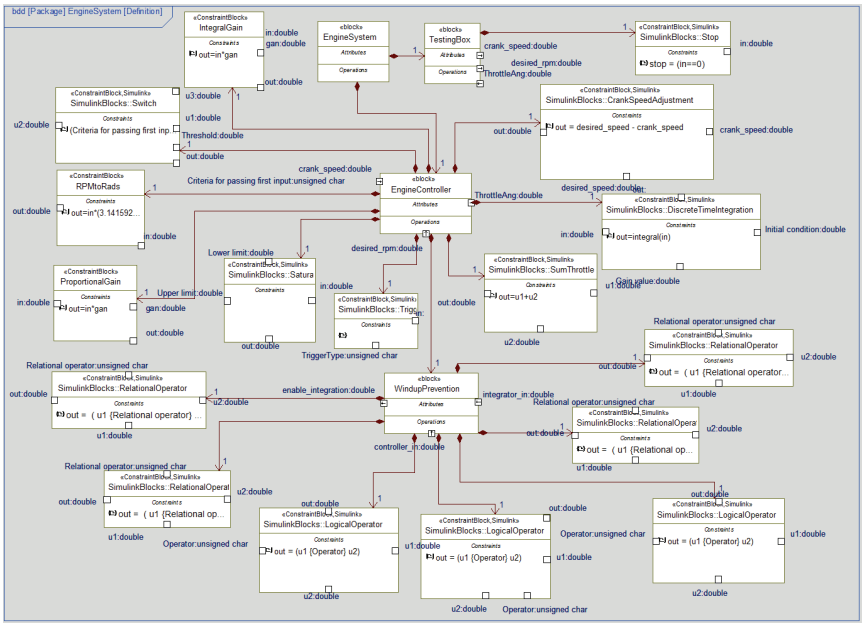

Fig. 5 Platform independent model (Rhapsody Model)

This platform independent model was then transformed into a Simulink model (still hardware independent at this stage), which would be customized later to become platform specific model for hardware specific deployment.

Fig. 6 shows the Simulink engine controller model transformed from the Rhapsody engine controller model, interacting with the vehicle plant system. IO connections are used within the model to allow interaction with the engine controller. These IO connections are realized by installing a NI-6024E PCI card in the xPC target PC. The NI PCI-6024E card used is a 200 kS/s, multifunction DAQ card with two 12-bit analog outputs; 8 digital I/O lines; and two 24-bit counters.

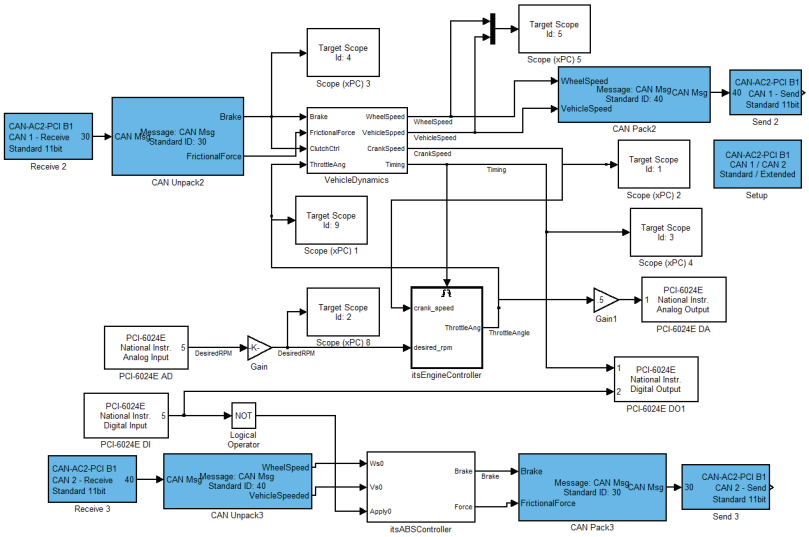

Fig. 6 Real time Platform independent model (Simulink) showing Vehicle Dynamics (Plant), Engine Controller and ABS Controller with CAN Communication

The figure also shows the ABS controller interacting with the vehicle plant system via CAN communication. The CANBUS system has been realized by installing a Softing CAN-AC2-PCI card in the xPC target PC. The Softing CAN-AC2-PCI is an active, two-channel CAN bus PCI interface card with the data transmission rate of up to 1MB/s.

B. Platform Specific Model

In order to transform the Platform Independent Model (PIM) of the engine Controller to the Platform Specific model (PSM) for the PIC24 microcontroller, it was necessary to establish the Microchip IO connections to prepare the model for the PIC microcontroller target. However, as the model was initially developed with PC control settings, it is necessary to modify the data type of the model as some of the data types are affected by the fact that the PIC24 microcontroller is 24-bit while the PC used is 32-bit.

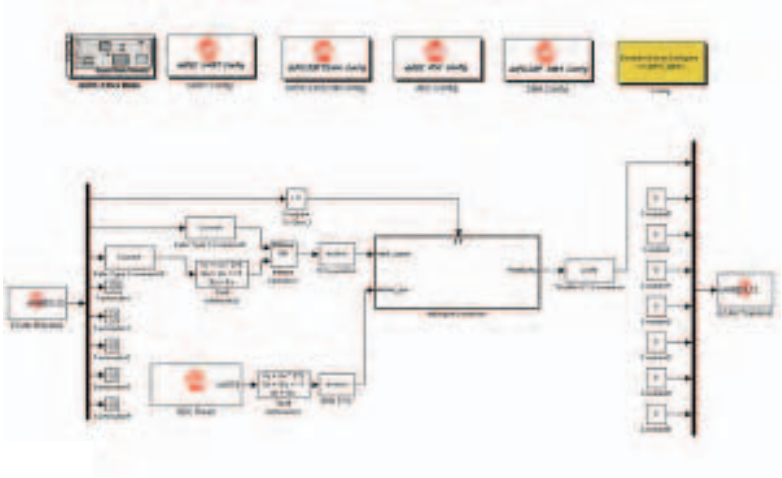

Fig. 7 Platform Specific Model (for PIC24 microcontroller)

However, these were the only modifications necessary, which demonstrated the significant savings in term of development time and effort. A snapshot of the platform specific engine controller model ready for deployment on the microcontroller is depicted in Fig. 7.

While code can be generated with the Simulink Realtime Workshop, the MPLAB Integrated Development Environment (IDE) of the PIC24 Microcontroller offers the same functionality to start Simulink at the background and managed the code generation process.

In order to demonstrate the flexibility and variety of the controllers supported, the project team decided to deploy a Platform Specific Model of the ABS Controller onto LabVIEW. This was carried out using the NI LabVIEW Simulation Interface Toolkit (SIT).

The toolkit provides a seamless integration between the Simulink model and LabVIEW real-time I/O for prototyping, deployment, and HIL simulation. It automatically generates LabVIEW code to interface with a Simulink module resulting in a flexible and easy-to-use user interface. This was achieved by generating a DLL for LabVIEW from the Simulink model with the Simulink Real-Time Workshop. Data acquisition, CAN, and FPGA I/O can also be added through a configuration-based dialog [15].

C. xPC based Hardware-in-the-Loop Simulation

Figure 8 shows the overall arrangement of HIL simulation using xPC by MATLAB. The physical arrangement involved a host computer connected to an xPC target using Ethernet connection. The Ethernet crossover cable was used for higher data throughput and longer distances between host and target computer; as compared to the alternative serial communication connection. The vehicle plant model from the host PC was deployed on the target PC to allow real time interaction with the engine and ABS controllers.

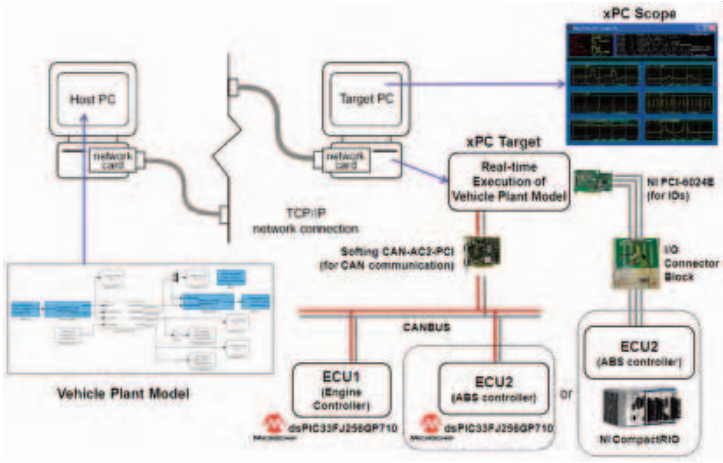

Fig. 8 Hardware-In-The-Loop Simulation using xPC by MATLAB

Fig. 9 shows the hardware in the loop simulation setup used by the project. The vehicle system model was running in the Target PC, with the two controllers interacting with the system via CANBUS communication. The ABS controller could later be implemented on the NI CompactRio and interacted with the system model via I/O connector block.

Such arrangements served to demonstrate the key concept of MDA approach where design from PIM can be implemented onto the controller of choice by transforming PIM to PSM, freeing the system developer from relying on specific technologies. Moreover, this arrangement demonstrated the key advantage of HIL simulation to allow deployment, test and verification of controllers without the need of costly real system hardware.

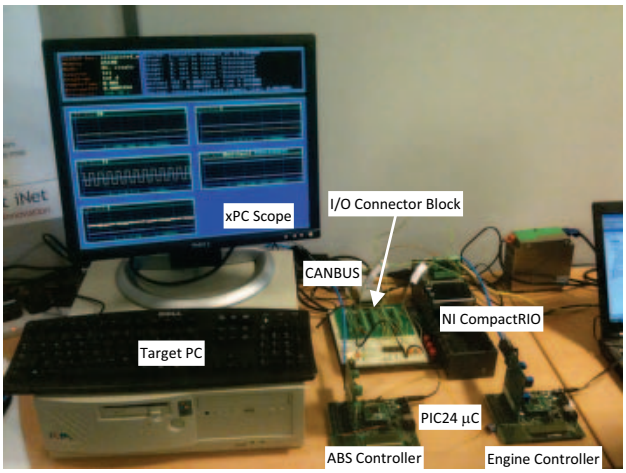

Fig. 9 HIL Simulation of the Vehicle Plant Model with Engine and ABS controllers

Fig. 10 shows the xPC Target scope output which provides a "window" into the main parameters (e.g. throttle angle, crank speed, timing, and wheel speed) of the vehicle system as a mean to verify that the controllers work successfully with the vehicle system.

Fig. 10 xPC Target scope output showing the main parameters of the vehicle system

V. FINDINGS

The project has demonstrated how model driven system engineering (MDSE) enabled by means of MDA approach HIL simulations can be used to support future vehicle designs by driving the system architecture from a platform independent perspective. The platform independent system architecture generated can then be migrated to a number of platform dependant solutions, which facilitate early system verification and testing, significantly reducing risk and development cost.

The MDA approach has enabled greater integration of applications and facilities across middleware boundaries, which is also applicable to other sectors. Companies adopting the MDA approach will become more agile in terms of incorporating sophisticated technology and thus adapt to technology changes throughout a product's lifecycle.

From a commercial perspective this will provide a competitive edge to companies adopting the MDA approach by facilitating early system verification and testing with subsequent risk reduction and compliance with stakeholder needs. The approach has increased the likelihood of on time delivery, helping to achieve greater efficiency and optimal system solutions, and providing future proofing against technology obsolescence.

Moreover, the model based design and graphical representation of the system facilitate important steps towards completely documented design. This is essential for preserving system knowledge and design, enabling easier reconfiguration and greater reuse. However, one drawback identified was the greater effort required to protect the system design from being infringed by third parties.

Industrial collaborators have shown interest to incorporate MDA in their design and development to fuller scale. Both the Mechatronics Research Centre and the Research School of Systems Engineering are committed to the Model Driven Architecture for their system developments.

Both universities have since introduced Model-based Design and Systems Engineering to both undergraduate and postgraduate students and their projects.

VI. CONCLUSIONS

The project has successfully outlined and applied next generation model driven system engineering techniques known as Model Driven Architectures (MDA), underpinned by hardware-in-the-Loop simulation, to allow the transport manufacturing sector to be more agile, competitive and responsive in providing technical solutions for future vehicle systems.

Suitable methodology and tools have also been established, supported by a newly identified and developed model transformation tool. These allow model driven system design and development to be carried out where the development of a train controller and braking system has been used as case study.

The project team is also confident that the research outputs from the project are equally applicable to the wider transport sectors which include aerospace, automotive and marine. Other sectors such as the manufacturing and automation will also benefit from the approach, to produce high level system architectures that are platform independent.

Companies adopting this approach are expected to be able to map their solutions onto other products much more easily – potentially opening up new market opportunities.

ACKNOWLEDGMENT

The authors would like to thank the Transport iNet, East Midlands Innovation (eminnovation.org.uk) of East Midlands Development Agency (EMDA) for their support and funding to enable this research to be carried out. The Transport iNet is part-financed by the European Union's European Regional Development Fund (ERDF). The Transport iNet was set up to allow individual sectors to work together, share ideas, technologies, intelligence, innovation and create a link between operational industries and the academic community.

REFERENCES

[1] Calic, T.; Dascalu, S.; Egbert, D., "Tools for MDA Software Development: Evaluation Criteria and Set of Desirable Features", *Fifth International Conference on Information Technology: New Generations, 2008*. ITNG 2008. 2008, Page(s): 44 – 50.

[2] Object Management Group (OMG), "MDA® Specifications". Available: http://www.omg.org/mda/specs.htm.

[3] Callowa, G., Kalawsky, R. and Watson, G., "Model Driven Architecture for Autonomous Systems – Modelling Representations", *5th SEAS DTC Technical Conference* - Edinburgh 2010.

[4] Model Driven Solutions and Data Access Technologies, "Model Driven Architecture (MDA)". Available: http://www.modeldriven.com/MDA.shtml, 2010.

[5] Broy, M., Feilkas, M., Herrmannsdoerfer, M., Merenda, S., Ratiu, D., "Seamless model-based development: From isolated tools to integrated model engineering environments", *Proceedings of the IEEE* 98(4), 526-545 (2010).

[6] Kalawsky, R. "Platform Independent Model Driven Architectures for Future Vehicle Systems". Available: http://syseng.lboro.ac.uk/mbse_flyer.pdf.

[7] Qamar, A., During, C. and Wikander, J., "Designing Mechatronic Systems, A Model-based Perspective, An Attempt to Achieve SysML-Matlab/Simulink Model Integration", *Proceedings of 2009 IEEE/ASME*, International Conference on Advanced Intelligent Mechatronics (AIM09). Singapore, 2009.

[8] Vanderperren, Y. and Dehaene, W., "From UML/SysML to Matlab/Simulink: Current State and Future Perspectives", *Proc. of DATE'06*, Munich 2006.

[9] Houghton, S. "The Ongoing Pursuit of an Integrated Toolset for Model-Based Design: Benefits and Challenges", SELEX Sensors and airborne Systems.

[10] Richard Boldt, R., "Combining the Power of MathWorks Simulink and Telelogic UML/SysML- based Rhapsody to Redefine the Model Driven Development Experience. Telelogic, 09 November 2007.

[11] Boldt, R., "Combining the Power of MathWorks Simulink and Telelogic UML/SysML- based Rhapsody to Redefine the Model Driven Development Experience", Telelogic, 09 November 2007.

[12] Opal-RT Technologies, Inc., "About Hardware in the Loop and Hardware in the loop Simulation". Available: http://www.opal-rt.com/about-hardware-loop-and-hardware-loop-simulation, 2011.

[13] Müller-Glaser, K., Reichmann, C. and Kuehl, M., "Supporting system level design of distributed electronic control units for automotive applications". Available: http://msdl.cs.mcgill.ca/people/mosterman/campam/hlsla07/index.html/mueller-glaser.pdf.

[14] Sjöstedt, C.J., Shi, J., Törngren, M., Servat, D., Chen, D., Ahlsten, V., Lönn, H., "Mapping Simulink to UML in the design of embedded systems: Investigating scenarios and transformations", *OMER4 Workshop: 4th Workshop on Object-oriented Modeling of Embedded Real-Time Systems*, 2007.

[15] MathWorks, "xPC Target 5 User's Guide". Available: http://www.mathworks.com/help/pdf_doc/xpc/xpc_target_ug.pdf.