# Towards an Architecture-Centric Approach dedicated to Model-Based Virtual Integration for Embedded Software Systems

Huafeng Yu[1], Jean-Pierre Talpin[2], Sandeep Shukla[3], Prachi Joshi[3], and Shinichi Shiraishi[1]

[1] TOYOTA InfoTechnology Center, U.S.A.
465 N Bernardo Avenue, Mountain View, CA 94043, U.S.A.
huafeng.yu@us.toyota-itc.com
[2] INRIA Rennes - Bretagne Atlantique,
Campus de Beaulieu, 263 Avenue Général Leclerc, 35042 Rennes, France
[3] Virginia Polytechnic Institute and State University
Falls Church Campus, 7054 Haycock Rd., Falls Church, VA 22043, USA

**Abstract.** Current embedded systems are increasingly more complex and heterogeneous, but they are expected to be more safe, reliable and adaptive. In consideration of all these aspects, their design is always a great challenge. Developing these systems with conventional design approaches and programming methods turns out to be difficult. In this paper, we mainly present the informative background and the general idea of an ongoing yet young research project, including the model-based design and an architecture-centric approach, to address previous challenges. Our idea adopts a formal-methods-based model integration approach, dedicated to architecture-centric virtual integration for embedded software systems, in an early design phase. We thus expect to improve and enhance *Correct By Construction* in the design. The considered formal methods consist of timing specification, design by contracts, and semantics interoperability for models to be integrated in the system. The application domains of our approach include automotive and avionic systems.

**Keywords:** Virtual integration, model-based design, AADL, timing specification, design by contract, semantics interoperability

## 1 Introduction

Current embedded systems are increasingly more complex and heterogeneous, but they are expected to be more safe, reliable and adaptive [8] [16]. In consideration of all these aspects, their design is always a great challenge. Complexity in the design and implementation is a common issue for current avionic and automotive systems. In the current system design, verification and validation (V&V) is also a key concern, particularly for safety-critical systems. These systems generally require great V&V effort to avoid unexpected system behavior.

Moreover, the design is expected to be validated as early as possible due to the huge cost of correction in the late-phase implementation. Design validation in an early phase has become one of the key solutions to reduce the overall V&V cost.

In this paper, we mainly present the informative background and the general idea of an ongoing yet young research project, including the model-based design and an architecture-centric approach, to address previous challenges. Our idea adopts an formal-methods-based model integration approach, dedicated to architecture-centric virtual integration for embedded software systems, in an early design phase. By applying formal methods in an early design phase, we expect to improve and enhance *correct by construction*. The formal methods to be considered consist of timing specification, design by contracts, and semantics interoperability for models to be integrated in the system. The application domain of our approach include avionic and automotive systems.

## 2 Research Challenges

High-level modeling has been widely adopted as a promising solution to address the system complexity issue [33]. High-level modeling languages, such as UML[27], SysML[1] and MARTE[26], have been widely adopted, thanks to its standardization for modeling. AUTOSAR[2] and EAST-ADL[9] are domain-specific languages for automotive systems. AADL[32] (Architecture Analysis and Design Language) is an SAE standard dedicated to architecture description and modeling for avionic and automotive systems. AADL provides an industry standard, textual and graphic notation with precise semantics to model applications and execution platforms and is supported by commercial and open source tool solutions—including Open Source AADL Tool Environment (OSATE) [28]. Matlab/Simulink[21] is a dataflow language for modeling, simulating and analyzing dynamic systems. Modelica[23] is an object-oriented modeling language for component-based complex systems. These high-level languages enables domain specific modeling and analysis of complex embedded systems. SCADE [12] is an integrated design environment dedicated to rigorous design of safety-critical systems[4].

#### Multi-paradigm modeling

All the languages mentioned previously are considered as candidate languages in high-level modeling for embedded systems. Multi-languages can be used in the same design because of system modeling from different views, for example, software, architecture, etc.; and different purposes, such as analysis, verification, and evaluation. Furthermore, different languages may adopt different formalism, e.g., state machines, dataflow, communicating sequential processes, differential equations, as backstage support. So the first challenge at the modeling language level is how to harmonize multiple paradigm modeling [24] [25] in the same

design, particularly, when we consider a reliable integration followed by using formal techniques for analysis and V&V at the system level.

**An avionic co-modeling example.** Co-modeling for the system-level design has been explored in [37] [36], where AADL was used to model the architecture part and Simulink was used to model the behavior part of an avionic case study, called simplified Airbus A350 doors management system. However, semantic difference of the two models makes the integration problematic. In order to have a clear and unambiguous integration, a formal model of computation (MoC), called Polychrony [17], was adopted as an intermediate model. This MoC is based on the synchronous/polychronous timing semantics. The later formal analysis, verification, and scheduling were mainly performed on the basis of the same MoC.

#### Integration frameworks

In Polychrony, the integration is performed at the polychronous MoC level[36]. Polychrony provides model transformations from AADL and Simulink (via GeneAuto[35]) to the polychronous MoC. In order to keep the semantics coherent, both AADL and Simulink models adopt the polychronous semantics. Based on the same polychronous semantics, the composed model can used for analysis, verification, and simulation or be translated into other formal models for formal verification and scheduling. So in this integration scheme, the core polychronous model provides formal semantics support and its environment provides tool connection. Model-based system integration has also been discussed in [34] with regard to cyber-physical systems, [6] for tool integration platform, [31] based on SOA (Service of Architecture), [10] for heterogeneous models integration, and [29] for real-time software engineering. AUTOSAR[2] aims at component-level integration for automotive systems. System Architecture Virtual Integration (SAVI) program [30] [13] aims at creating an architecture-centric model repository to support analysis of virtually integrated system models related to performance, safety, and reliability, and so on. It also enables to discover system-level faults at the early design phase, thus reduce risk, cost, and development time.

## 3 A Model-Based Architecture-Centric Virtual Integration Framework

Based on the previous exploration of design issues and the state of the art of solutions in research, we find an architecture-centric model-based integration framework is necessary for the next-generation design of automotive software systems. The framework is expected to provide the following advantages: reliable model integration, fast and early-phase design validation, architecture optimization enabling, easy access to current matured software development tools and environment, etc. With this objective in mind, we first propose a model-based

architecture-centric virtual integration approach, in the framework of model-based systems engineering [11], for the design of next-generation automotive systems. This approach is involved in mostly *correct by construction* technologies, rather than a posteriori *Verification & Validation* in the implementation phase. We adopt different modeling languages with regards to different views of the system, for example, AADL for architecture modeling and Simulink for behavioral modeling, etc. The main research topics in the project include: timing specification [5], design by contracts and semantics interoperability for the purpose of a reliable model integration, which are explained in the following subsections.

### Timing specification

With all the concerns in the embedded system design, timing is one of the most significant ones. In general, the timing issue becomes more explicit when architecture is considered and the system is integrated, due to the gap between software and architecture design. In our project, we consider high-level, formalized timing constraints to be defined, observed and analyzed based on software architecture, specified in AADL. From this point of view, an architecture centric approach is adopted for the model integration in our project. Considering abstraction in the system design, we advocate the modeling of synchrony and time as software and hardware events, which are related to synchronization in an architecture specification. Compared to real time, synchronous logical time, applied on both software and architecture, provides an algebraic framework in which both event-driven and time-triggered execution policies can be specified.

In the framework of our project, we define the semantics and algebra with regard to logical timing constraints and specification, and support the submission of a timing-related annex to the SAE standard AADL[32]. This annex will define a synchronous and timed specification framework to formally model time domains pertaining to the design of embedded architectures, including the specifications of automotive software architectures. The behavior annex of AADL are considered as the vehicle to implement this model, together with a timing annex (TA), as a mean to represent abstractions of these behavior annexes using clock constraints and regular expressions.

### Design by contract

*Design by contract* [22] [15] is also adopted in our approach in the project. Contracts play a significant role in the safe and reliable model integration in our approach. We first analyze high-level requirements from automotive or avionic systems, from which *formalizable* requirements are then extracted according to the technical formalizability and verifiability. These requirements are expressed in formal languages so that they can be used to build the contracts for the integration of models that implement corresponding functionality. The contracts are expected to consider different criteria for safety, performance, cost, timing constraints, and so on. A mathematical framework will then be built to define the

composition of these models, together with the contracts on them, in a formal way. The contracts and their associated models will be checked by modeling checking technologies [14] [19] [7] .

**Semantics interoperability**

One of the main issues in the composition of models is semantics difference between heterogeneous models and different formalism. One of the feasible solutions to this issue is to have a common model as the intermediate formal model, and all other models are translated into the common model. An example can be found in [37]. The intermediate model provides the formal semantics, based on which, expected properties of the original models and their integration are checked. However, this requires a semantics preservation in the model translation, which is not practical in most cases. Another solution is related to formal semantics interoperability. Some work can be found in [3] [20], [18]. Our current research topic is focusing on the study of differences between the models, which can lead to issues in the model translations, from the point of view of model semantics, particularly timing semantics and operational semantics. The expected result of this research is intended to provide a foundation of the previous two research topics.

## 4 Conclusion

In this position paper, we have presented several important issues in current system design related to embedded systems, such as multi-paradigm modeling, integration framework, and formal semantics issues. A brief survey of corresponding research topics was also presented. We, hence, propose a model-based architecture-centric integration approach, considering timing specification, design by contract and semantics interoperability as main topics of research. Based on these research, a model-based integration framework is expected to be built, which is dedicated to model-based systems engineering for next-generation automotive systems.

## Acknowledgment

The authors appreciate the valuable advices from Ryo Ito and Kazuhiro Kajio (Toyota Motor Corporation).

## References

1. Systems Modeling Language (SysML). http://www.sysml.org/specs.
2. AUTOSAR (AUTomotive Open System ARchitecture). http://www.autosar.org/.
3. A. Benveniste, B. Caillaud, L.P. Carloni, P. Caspi, and A.L. Sangiovanni-Vincentelli. Composing Heterogeneous Reactive Systems. *ACM Transactions on Embedded Computing Systems*, 7(4), 2008.

4. A. Benveniste, P. Caspi, S. Edwards, N. Halbwachs, P. Le Guernic, and R. de Simone. The Synchronous Languages Twelve Years Later. *Proceedings of the IEEE*, 2003.
5. L. Besnard, E. Borde, P. Dissaux, T. Gautier, P. Le Guernic, and J.-P. Talpin. Logically timed specifications in the aadl : a synchronous model of computation and communication (recommendations to the sae committee on aadl. Technical Report 446, INRIA, 2014.
6. M. Broy, M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu. Seamless Model-Based Development: From Isolated Tools to Integrated Model Engineering Environments. *Proceedings of the IEEE*, 98:526–545, 2010.
7. Darren Cofer, Andrew Gacek, Steven Miller, Michael W Whalen, Brian LaValley, and Lui Sha. Compositional Verification of Architectural Models. In *NASA Formal Methods*, 2012.
8. DARPA. Adaptive Vehicle Make (AVM) Project. http://www.darpa.mil/Our_Work/TTO/Programs.
9. EAST-ADL. http://www.east-adl.info.
10. J. Eker, J.W. Janneck, E.A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming Heterogeneity - the Ptolemy Approach. *Proceedings of the IEEE*, 91(1):127–144, 2003.
11. J.A. Estefan. Survey of Model-Based Systems Engineering (MBSE) Methodologies. Technical report, INCOSE MBSE Initiative, 2008.
12. Esterel Technologies. SCADE Suite. http://www.esterel-technologies.com/products/scade-suite/.
13. P. Feiler, J. Hansson, D. de Niz, and L. Wrage. System Architecture Virtual Integration: An Industrial Case Study. Technical report, Software Engineering Institute, Nov. 2009. CMU/SEI-2009-TR-017.
14. A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A Tool for Automatic Verification of Probabilistic Systems. In *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'06, pages 441–444, Berlin, Heidelberg, 2006. Springer-Verlag.
15. J.-M. Jézéquel and B. Meyer. Design by Contract: The Lessons of Ariane. *Computer*, 30:129–130, 1997.
16. Xiaoqing Jin, Jyotirmoy Deshmukh, James Kapinski, Koichi Ueda, and Ken Butts. Challenges of Applying Formal Methods to Automotive Control Systems. In *NSF National Workshop on Transportation Cyber-Physical Systems*, 2014.
17. P. Le Guernic, J.-P. Talpin, and J.-C. Le Lann. Polychrony for System Design. *Journal for Circuits, Systems and Computers*, 12:261–304, 2002.
18. E. A. Lee and A. Sangiovanni-Vincentelli. A Framework for Comparing Models of Computation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1217–1229, 2006.
19. A. Legay, B. Delahaye, and S. Bensalem. Statistical model checking: An overview. In *Runtime Verification*, 2010.
20. D. Mathaikutty, H. Patel, S. Shukla, and A. Jantsch. Modelling Environment for Heterogeneous Systems based on MoCs. In *Forum on specification and Design Languages (FDL)*, pages 291–303, 2005.
21. MathWorks. The MathWorks: Matlab/Simulink. http://www.mathworks.com/products/simulink/.
22. B. Meyer. Applying 'design by contract'. *Computer*, 25(10):40–51, Oct 1992.
23. Modelica and the Modelica Association. https://www.modelica.org.

24. P. J. Mosterman and H. Vangheluwe. Computer automated multi-paradigm modeling: An introduction. *SIMULATION: Transactions of the Society for Modeling and Simulation International*, 80(9):433–450, 2004.
25. K.D. Müller-Glaser, G. Frick, E. Sax, and M. Kühl. Multiparadigm Modeling in Embedded Systems Design. *IEEE Transactions on Control Systems Technology*, 12(2):279–292, 2004.
26. Object Management Group (OMG). The UML Profile for MARTE: Modeling and Analysis of Real-Time and Embedded Systems. http://www.omg.org/spec/MARTE/1.1/PDF, June 2011.
27. OMG. Unified modeling language (uml). www.uml.org/.
28. OSATE. OSATE V2 Project. https://wiki.sei.cmu.edu/aadl/index.php/Osate_2.
29. Maxime Perrotin, Eric Conquet, Julien Delange, André Schiele, and Thanassis Tsiodras. TASTE: A Real-Time Software Engineering Tool-Chain Overview, Status, and Future. In *SDL 2011: Integrating System and Software Modeling*, 2012. Lecture Notes in Computer Science Volume 7083, pp 26-37.
30. D. Redman, D. Ward, J. Chilenski, and G. Pollari. Virtual integration for improved system design,. In *The First Analytic Virtual Integration of Cyber-Physical Systems Workshop in conjunction with RTSS*, 2010.
31. A. Rossignol. The Reference Technology Platform. In *CESAR - Cost-efficient Methods and Processes for Safety-relevant Embedded Systems*. Springer, 2013.
32. SAE Aerospace (Society of Automotive Engineers). Aerospace Standard AS5506A: Architecture Analysis and Design Language (AADL) . *SAE AS5506A*, 2009.
33. D.C. Schmidt. Model-Driven Engineering. *IEEE Computer*, 39:25–31, 2006.
34. J. Sztipanovits, X. D. Koutsoukos, G. Karsai, N. Kottenstette, P.J. Antsaklis, V. Gupta, B. Goodwine, J.S. Baras, and S. Wang. Toward a Science of Cyber-Physical System Integration. *Proceedings of the IEEE*, 100(1):29–44, 2012.
35. A. Toom, T. Naks, M. Pantel, M. Gandriau, and I. Wati. Gene-Auto: An Automatic Code Generator for a Safe Subset of SimuLink/StateFlow and Scicos. In *European Conference on Embedded Real-Time Software (ERTS'08)*, 2008.
36. H. Yu, Y. Ma, T. Gautier, L. Besnard, J.-P. Talpin, and P. Le Guernic. Polychronous Modeling, Analysis, Verification and Simulation for Timed Software Architectures. *Journal of Systems Architecture (JSA)*, 59(10):1157–1170, 2013.
37. H. Yu, Y. Ma, Y. Glouche, J.-P. Talpin, L. Besnard, T. Gautier, P. Le Guernic, A. Toom, and O. Laurent. System-level Co-simulation of Integrated Avionics Using Polychrony. In *ACM Symposium on Applied Computing (SAC'11)*, 2011.