Position Paper: Need for Architecture Description Language with Standardized Well Defined Meaning for Architecture Centric Engineering of Cyber-Physical Systems

Bruce Lewis, US Army AMRDEC

Abstract: System complexity, especially from the perspective of the dynamics of system interaction, is rapidly accelerating as computers are used to integrate applications at each level of system execution, from subcomponents to systems to systems of systems. This new complexity is expressing itself in the cost of system integration and the issues of reliability, dependability and safety, as well as overall system performance. To avoid very significant costs and risks to the program, we must virtually integrate these systems before building them, which is not easy to do with all the complexities of computer system interaction. In addition, due to the complexity of large systems and the lack of funding to adequately test them, we must enhance reliability through additional analytical and formal verifications. We must understand these systems from a consistent model, integrating lower level models with perhaps domain specific languages of specification, into an analytical architectural framework. At any time during the lifecycle of the system, this architectural model reflects the current state of system development. From it, we drive many forms of analysis to determine architectural compliance to system constraints and requirements. This system model must be architectural and component based if we are to understand the interactions, impact of change, and the emergent properties. The Architecture Analysis & Design Language (AADL) was designed for this purpose and is being applied in the area of software reliant system development. Architectural analyses for Cyber-Physical systems will add additional complexities to the expression and analysis of systems. To be effective in addressing these systems for analysis, a similar capability will be required. Cyber-Physical system models and analyses themselves will not integrate or be consistent without being formed against a common, standardized, well defined architecture description language for understanding compositional effects, cross contractor integration, incremental development and multi-dimensional analysis. Hence the use of such a language is critical to the goal of Cyber-Physical virtual integration.

Where Are We Today

Current industry practice is primarily based on a “build then test then fix the emergent issues then test then fix...”. This has served us up to this point, especially with the help of very senior, very smart people who can find the issues that surface during integration. However, this brings up the issue of what testing is not finding in these complex systems as well as the cost and schedule impact of making the system work. The issue of cost of integration of complex systems is being recognized especially in the aviation industry. The scale and heterogeneous nature of Cyber-Physical systems seem even more complex.
The aviation industry is experiencing a growth in software cost and complexity that threatens to overrun the cost of the full system development. The System Architecture Virtual Integration (SAVI) [1] project of the Aerospace Vehicle Systems Institute developed data that shows that for the next generation aircraft development, software cost will exceed $10 Billion. This threatens the development feasibility of these systems. SAVI’s approach to reduce these costs, an industrial consortium, led by Boeing and including Airbus, Lockheed Martin, BAE Systems, Rockwell, Goodrich, the FAA, Army, and NASA, is virtual architecture integration and analysis. The reason relates to the high cost of integration due to the late discovery of system integration issues. Figure 1 illustrates the potential cost reduction impact of error discovery before system integration. The chart shows that 70% of the errors are created in the requirements and design phases, but only 3.5% are found during these phases. This indicates a significant weakness in the current process. Currently about 80% of the errors are found during and after system integration. The multipliers of 16x, 40x and 110x are cost multipliers based on the phase of discovery of the errors. If we find the error during the requirements/design phase, the cost of rework for that error is reduced by these factors. There is significant opportunity for error reduction in system integration through a virtual, predictive hardware/software integration of the architecture. Cyber-Physical systems virtual integration will further broaden our ability to find issues in the system. The published SAVI report [1] from phase 1 is available. The current SAVI demo, phase II, is looking at Cyber-Physical interactions in the aviation system.

Figure 1: Only 3.5% of errors found in requirements and design, 80% found during or after integration test

In order to understand architectural issues, the behavior of system interaction must be captured in the architecture and analysis applied to assess critical qualities required of the architecture for system
Today’s systems are highly software reliant, so the interactions across the system occur through the software and computer hardware, adding significant complexity to system modeling, analyses, and simulation. These analyses must be integrated in the architecture context so that the impact of the component or change in any component can be evaluated across the system model. In addition the architectural model must be incrementally developed by the integrator and his suppliers, based on the components with sufficient meaning to understand their interaction dynamics in the context of system execution. The system architecture specification must be consistent, including semantics, data and properties that drive the models, not just linked models from multiple teams. The representation of the architecture must be machine process-able with a standard, well formed set of semantics supporting engineering and formal analysis. In this context, data for multiple analyses can be extracted from the architecture specification residing in a shared system repository. See figure 2. Through the architectural context and static and dynamic semantics, missing information can be detected and analyses can be based on the level of completion of the architecture.

Figure 2 Extracting analysis parameters from the architecture model.

Now the designer is in a position to evaluate change in a specific architectural property against an architectural requirement and more powerfully to evaluate the ripple impact of a changed property or component across multiple critical qualities at multiple levels in the architecture. Without well defined semantics in the architecture description language, designers need to add semantics, or override semantics or try to integrate semantics across multiple tools and models, in effect creating a custom
language. This is a significant task, especially when integration of models and analysis is needed across multiple forms of analysis and across multiple contractors.

The Architecture Analysis & Design Language [2] is a language with the well formed semantics to capture the architecture for both static and dynamic analysis. It was developed for this purpose as well as to support the generative integration of the components of the system to the specification supported by its analysis. The language provides a standard core set of capabilities and then enables extension through property sets and annexes. The annex mechanism provides a means for extension for Cyber-Physical systems. Language semantics support both engineering analysis and formal analysis as demonstrated by the number of analysis tools being developed for the language using its semantics. The report “Software Reliant System Qualification” [3] provides an overview of analysis methods and tools across multiple domains that have employed this standardized language for its semantic tightness and architectural framework. See papers [6-9] as additional examples related to system verification and validation. It has enabled high efficiency and high integrity automated system integration [4, 10]. NASA has developed an IV&V method using it [5].

Such a standardized language is needed for the challenge presented by the additional analytical context presented by Cyber-Physical Systems. The AADL committee is exploring possible annexes to better support Cyber-Physical Systems and SAVI is experimenting with the AADL in the Cyber-Physical context.

References


