A Pervasive Approach to Software Validation in Adaptive Cyber-Physical Systems

Matthew B. Dwyer, Sebastian Elbaum, Steve Goddard, Gregg Rothermel
Department of Computer Science and Engineering
University of Nebraska - Lincoln
Lincoln, Nebraska
{dwyer,elbaum,goddard,grother}@cse.unl.edu

September 8, 2006

1 Position
Future cyber-physical systems are the vanguard of a new breed of software that must face opposing forces demanding high confidence and high adaptivity in systems. Cyber-physical systems of concern are systems of computer systems that interact with the physical world in real-time. These systems must sense constraints imposed by a dynamically changing environment and predictably react to those changes in real-time.

Cyber-physical systems exist today, but in a much smaller scale in size and complexity than the anticipated cyber-physical systems of the future. Consider, for example, future aviation systems. Advances in autonomous systems will enable unmanned flight to become commonplace and will provide the opportunity to offload the monitoring and planning of an increasingly dense airspace from human pilots and controllers. Autonomous software, by its very nature, is highly adaptive — it must assess the system environment, plan an appropriate response, and execute that plan. It must do this correctly even in the face of unforeseen circumstances in the environment or failures on-board the system it is controlling. Assuring high-confidence in such systems goes well beyond the state of the art in software systems development and testing.

Our position is that researchers, and software and systems developers, especially those in aerospace and avionics, must adopt a pervasive approach to software validation and monitoring; an approach in which validation continues into the field with deployed systems. Moreover, pervasive validation must be grounded in meaningful, demonstrable, and formally defined evidence of software quality. Evidence may be contributed to the software quality case for a system through the entire software development process, from early-phase formal requirements modeling and reasoning through pre-deployment static analysis, verification and testing activities to assess behavioral conformance with requirements. It may also be contributed through the entire software lifetime — that is, over the sequence of releases that constitute an evolving software system. Finally, evidence must be gathered while the system is deployed in the field for quality requirements that cannot be assured statically. An integrated body of such evidence would provide a basis for software certification and acceptance, and re-certification following system modification. Simultaneously, this evidence will allow quality-oriented autonomic adaptation of systems to steer their run-time behavior to maximize system safety and meet mission objectives.

2 Fundamental Limitations
Our position that a pervasive approach to software validation will be needed is predicated on the fact that the complexity of today's systems is already beyond the state of the art in software verification and validation. Consider, for example, the following fundamental limitations.

Complex physical environments. Cyber-physical systems will operate adaptively based on environment stimuli and the specific mission of the system. Research on control-theoretic approaches to adapt system operations to the environment is advancing rapidly, but there are limits to what can be achieved. Current V&V techniques are ill-equipped to handle the complexity of cyber-physical systems environments, e.g., it is impossible to cover them in the usual sense of testing, and, consequently, it is doubtful that they will effectively complement ongoing research to provide a complete solution to system validation.

Remote upgrades. Cyber-physical systems are deeply embedded and they will evolve in place. This means that upgrades and fixes will be deployed remotely. Current V&V techniques rely almost exclusively on pre-deployment analysis and testing. This fundamental approach to V&V is contrary to the concept of cyber-physical systems.

A multiplicity of cyber-physical system instances. There may be a large variety of systems that share common components. When a component gets upgraded it may be difficult to know all of the differ-
3 Research Challenges

Software for future cyber-physical systems will include advanced in software-enabled control and technologies supporting distributed systems-of-systems. Rather than focus on the challenges associated with specific technical advances in those areas, we believe that it is crucial to emphasize the systemic and cross-cutting challenges related to system validation that developers of such systems will face.

Challenge 1: Develop techniques for precisely characterizing system, software and environment behavior. A precise characterization of the physical environment and the hardware’s sensing of that environment, is essential for describing the inputs to and corresponding behavior of system software.

Challenge 2: Define a total lifetime software validation process that produces an explicit body of evidence of software product quality. Software products throughout the life-cycle must be validated and associated with quality measures to enable traceability of linkages between assurances of quality in early phase artifacts, such as requirements models, and late phase artifacts, such as deployed code. The same use of validation and measures must also transcend single “releases”, enabling the assessment of systems as they evolve, and as they utilize components or subsystems that evolve.

Challenge 3: Develop adaptive economic models that relate the costs of validation strategies to the benefits of reducing various forms of risk, and that drive the selection of strategies on particular systems. Current assessments of validation strategies have been based largely on simple cost and benefit measures, paying little attention to context and to the factors under which costs and benefits vary. These assessments can’t scale effectively to cyber-physical systems in which cost-benefit tradeoffs will vary widely across settings. Furthermore, because risks and benefits may also vary across time with these systems, models that adapt, given the provision of additional data through monitoring, are required.

Challenge 4: Develop efficient adaptive approaches for performing online monitoring and validation activities of deployed systems to cope with the dynamic component upgrades and nature of a cyber-physical system’s environment. Specific challenges involve the adaptation of model-based control concepts to produce autonomic systems that drive the selective execution of online validation activities based on mission-specific economic models. In support of this, it essential that technologies for monitoring run-time behavior be developed that are effectively transparent, in terms of resource consumption, from the perspective of mission-critical software.

4 Promising Innovations and Abstractions

The challenges identified are formidable. If we are to meet these challenges, a bold new research agenda must be set that creates new innovations and abstractions. The following ideas provide a starting point.

Online Validation. The use of sophisticated online validation has been rejected by most system developers due to concerns about significant impacts on execution time. We believe that recent innovations in low-cost and removable software monitoring and flexible scheduling techniques can be used to provide cost-effective online monitoring.

Seamless Validation. All validation activities are oriented towards the same goal: demonstrating correct software and systems operation. Traditionally, there has been a disconnect between offline, i.e., static, and online, i.e., dynamic, techniques and they are thought to be incompatible in certain fundamental ways. We believe that viewing them as elements of a single spectrum of techniques, integrated through explicit evidence of correctness provides a powerful abstraction to leverage for building new validation technologies.

Model-based Adaptive Validation. Researchers have used model-based control concepts to adapt the performance of software in response to sensed system inputs. We believe that the same concepts can be effectively applied to adapting both the process of validating software before deployment as well as online during system execution. Innovative economic models tuned to a system’s mission objectives will help prioritize pre-deployment activities and will provide online model-based control to target the monitoring of system functionality to address changing mission modes and environment interactions. For example, an upgraded component will come with a certificate describing the extent to which it has been validated. The model will incorporate this information and use it to decide the contexts in which the component may be used reliably or the monitoring of component behavior that is needed to safeguard against erroneous behavior.

A System Safety Net. Even if there were a perfect method for synthesizing a correct implementation from a specification it would still not solve the problem of developing trustworthy cyber-physical systems. Humans will always drive the development of requirements and specifications and they may well make errors in defining proper system behavior or, more subtly, in characterizing the nature of the environment in which...
the system will operate. To guard against such errors, reliable cyber-physical systems must operate with an online safety net that executes transparently with respect to system functional and temporal requirements and is able to adapt as system execution proceeds. This is a powerful and broad abstraction of the inter-related benefits of multiple technologies.

Integrated Models of Software Behavior. Existing measures of software quality are an incomplete patchwork of approaches that provide only the weakest information about how a program may operate. We must define approaches to capture a wide-variety of disparate forms of information about software behavior and to integrate that information into a holistic view of software product quality. For example, rich notions of behavior coverage are required to support claims of correct operation across a wide range of system execution contexts. Such notions must account for the space of possible program executions rather than sampling specific syntactic criteria of a program, such as MCDC.

Validating Intent. Today’s systems are specified and validated using very low-level actions. Next-generation cyber-physical systems will require the specification of intended action that can be observed, measured, and monitored in both offline, e.g., static analysis, and online, e.g., dynamic analysis, modes. Unless new, high-levels of specification abstractions can be formalized and validated, the complexity of future systems will be limited by our ability to specify them. Imagine trying to develop an enterprise-level application using assembly language. The evolution of development languages has dramatically increased the complexity of the systems we are able to build and deploy. A similar evolution is needed in the way we specify and validate systems.

5 Roadmap

There are four key strands of research that need to be pursued: (1) economic model development that incorporates aspects of cost/risk appropriate for the cyber-physical system domain, (2) low-level run-time transparent monitoring capabilities, (3) adaptive, model-based control to target V&V activities, and (4) integrated models of behavior and technologies that produce V&V “results” in the form of those models. These activities, especially item (1), clearly need systems and software engineering researchers working together. We envision a spiral roadmap, where each of these activities is explored in isolation, to a certain extent, and then integrations are investigated. Following this approach, we propose the following key milestones.

Years 1-7 Parallel investigation of activities (1)-(4).

While these activities should be done in parallel, they still require cross-disciplinary research teams.

Years 3-7 Investigate the integration of activity (4) with existing technologies, followed by the integration of new online, transparent monitoring technology developed in activity (2).

Years 6-10 Create an online autonomic quality capability by combining transparent monitoring with adaptive model-based control of V&V activities.

6 Conclusion

Large-scale, deeply embedded cyber-physical software systems create significant new research challenges. The software and system development and validation approaches of today are inadequate to meet the needs of next-generation cyber-physical systems. New research will be needed to meet these new challenges. It is our position that a new pervasive approach to software validation will be needed. This pervasive approach will generate an integrated body of evidence of software quality that provides a basis for software certification, or re-certification following system modification. Moreover, this approach will lead to quality-oriented autonomic adaptation of systems to steer their run-time behavior such that a system safety net is created while the platform meets mission objectives.

7 Biographies

Matthew Dwyer, the Henson Chair of Software Engineering at UNL, has over a decade of experience developing static software analysis techniques and tools and studying their application in practice. Sebastian Elbaum is an Associate Professor in CSE at UNL with expertise in run-time monitoring of deployed software and empirical studies of software testing techniques. Steve Goddard, an Associate Professor in CSE and College of Engineering Distinguished Scholar at UNL, is a highly-regarded expert in real-time and embedded systems and enterprise-scale decision support software. Gregg Rothermel, the Jensen Chair of Software Engineering at UNL, has expertise in program analysis, software testing, validation of evolving systems, economic models, and empirical studies. This team of researchers has led more than 35 research projects for ARO, DARPA, NASA, USDA, and NSF that have produced more than 13 software systems which have been leveraged for industrial use.