Introduction

BAE Systems has an established track record in the production of cyber-physical systems. The Advanced Information Technologies (AIT) division of BAE Systems has a history of collaboration with BAE business units that focus on the production of physical systems. Our contribution is centered on cutting edge technologies for building software engineering tools and infrastructure to enable the design, development, and sustainability of long-lived complex software-intensive systems, both in and outside of the transportation domain. AIT has sustained and nurtured this capability through collaborative work with university partners, including MIT and Vanderbilt University. In particular, we have a long term commitment to and focus on:

1. Model-based methods to provide higher levels of abstraction, correctness and automated code generation to increase the productivity of system developers,
2. Adaptation at design, load and run time. Toolsets and application models need to adapt at design time to changing requirements and evolving perception of the problem domain. Cyber-physical systems need to self-reconfigure at load time based on current conditions and requirements. Finally, at run time systems need to respond both to unforeseen conditions, and dynamic changes in requirements.
3. Correctness in the face of distributed systems, networked systems, upgrades, and adaptable systems.
4. Interdisciplinary research and transition from academic research to deployment: We combine research and leading technology from Artificial Intelligence (planning, scheduling, search, reasoning under uncertainty), computer science and software engineering (model based methods, formal verification, self adaptive systems), control theory (stability, closed loop control).

One of our current projects that is particularly relevant is called *Producible Adaptive Model Based Software (PAMS)* and is funded under DARPA's *Disruptive Manufacturing Technologies (DMT)* program. PAMS is focused on developing technologies to improve software producibility for large systems. As part of the PAMS effort we are applying next generation model-based software engineering methods to two ongoing cyber-physical programs at BAE, a software defined radio project, and the flight control systems for a military cargo jet. Our work under PAMS has demonstrated the benefits of software engineering tool support for agility and adaptation when designing and maintaining cyber-physical systems, and we propose to share our experience in this area at the national workshop.

Topic: Grand Challenges for transportation Cyber-Physical Systems

It is well known that the proportion of requirements for complex cyber physical systems that derive from software requirements has been steadily increasing and that the overall complexity of these systems has also been dramatically increasing. Software implementations of functionality that could also be realized in hardware provide flexibility and upgradability, but should not necessarily be viewed as less expensive. Adding another few thousand lines of code to the onboard functionality of a CPS might seem essentially free, with the fixed cost of the development amortized over many systems

and many years. However, the additional complexity does come at a substantial cost, especially if the cost of certification of safety critical software is taken into account. Transportation and medical cyber-physical systems, by their nature, are safety-critical, and the current best practices for the development and certification of safety-critical systems are not able to keep up with the desire to add networks, distribution, and adaptation to cyber-physical systems. This results in the following challenges facing transportation and other CPS with safety-critical aspects:

Challenge 1: Understanding the importance of software in CPS. There is a history of focusing on the physical aspects of CPS and underestimating both the importance and also the cost and complexity of the "cyber" component of CPS. Because it is harder to "kick the tires" so-to-speak of software, the initial prototyping and engineering effort tends to focus on the physical dimensions, components, fabrication, and specifications of the hardware elements of a CPS. Unless the hardware and software elements are specified, prototyped, and developed in unison, CPS as a whole will be plagued by software cost overruns, defects, and vulnerabilities.

Challenge 2: Developing tools and processes for certifying safety-critical adaptive, networked cyber-physical systems.

Building a "safety case" for a system must be supported by tools and processes throughout a CPS lifecycle – from requirements solicitation through maintenance. Furthermore, tools and processes must directly address the added complexity caused by adaptivity to the environment. Formal methods must be accessible to domain experts rather than limited to experts in mathematical logic. Heavyweight, time intensive methods that assume a closed world are inadequate for dealing with adaptive software in a dynamic world. As is well known in the area of adaptive flight control systems, but will become a broader issue as adaptive CPS proliferate, proving stability of an adaptive system is quite difficult; self-stabilization of adaptive systems is a fundamental challenge.

Challenge 3: Trading off software complexity and safety

While tools and processes to enable certification of safety critical CPS are slowly maturing, designers of CPS must directly address the trade-offs between feature-rich but un-trusted systems and more constrained (possibly less optimal) but dependable systems.

Topic: Architectures for common and specific Cyber-Physical Systems

We feel that there is a set of domain independent tools and processes that can be broadly useful to the development and deployment of future CPS. The following are important elements common to future CPS architectures:

- **Domain-specific models and languages developed using domain-independent technologies**: In both automotive and aerospace domains, the use of domain specific modeling tools and methodologies has proven useful for of the following reasons:

  - **Familiarity**: enabling domain experts to participate in the design and implementation of transportation CPS allows the direct encoding of domain expertise rather than having to also master software engineering skills.
  - **Productivity**: Models are likely at a higher level of abstraction than that presented by traditional general purpose programming languages or modeling tools, which also increases productivity.
  - **Correctness by construction:** By limiting the operations available to the modeler, the possibility for incorrect or infeasible systems is reduced.
  - **Correctness by analysis**: Domain-specific models are considered more amenable to analysis than code or models in general purpose languages.

- **Model-based Adaptation:** Models become the primary artifact for the construction of complex software systems. Code generators and interpreters operate on platform independent models to realize platform specific implementations. Integration of models, as well as reasoning about the models, into the runtime system, and development of model-based adaptive software, provides a basis for domain independent architectures for adaptive CPS.

Biographical Sketches and Contact Information

Gregory T. Sullivan, Ph.D. gregory.sullivan@baesystems.com (781)262-4553

Dr. Gregory Sullivan is a Principal Engineer at BAE Systems Advanced Information Technologies, specializing in model-based mixed initiative planning, model-based software engineering, adaptive systems, and dynamic programming languages. Dr. Sullivan received his Ph.D. in computer science from Northeastern University in 1996, having done research on formally correct program transformations. While a researcher at MIT's Computer Science and Artificial Intelligence Laboratory, Dr. Sullivan was funded by both NASA and DARPA and performed research on model-based programming and verification of autonomous systems, aspect-oriented programming, and dynamic programming languages and optimization. While at BAE Systems, Dr. Sullivan has been principal investigator on the DARPA JAGUAR program doing mixed initiative model-based planning, and is currently leading AIT's effort on the Software Producibility thrust of DARPA's Disruptive Manufacturing Technologies (DMT) program.

Basil Krikeles, Ph.D. basil.krikeles@baesystems.com (781) 262-4235

As Chief Architect at BAE Systems AIT, Dr. Krikeles has been involved in many challenging and innovative programs from proposal to completion. His experience includes concept development, customer interaction, project management, team leadership, and software architecture, development and process. Dr. Krikeles has over 15 years experience in the theory and practice of software development. His interests include: object-oriented software development, large scale software development, distributed applications, distributed object architectures, software producibility, semantic interoperability, and model-driven software construction. He is currently working on the Producible Adaptible Model-based Software (PAMS) project for DARPA's Disruptive Manufacturing Technologies program. Dr. Krikeles has more than twenty publications in Journals and Conference proceedings.