Discussion 1: Impediments to the Use of Verification in Industry

- design artifacts can't be easily connected to the verification representations/tools
- need to capture designs in an analyzable form
- behavior isn't clear from the design—difficult to infer a mathematical model
- black box is being put around the IP
- confusing about what the requirements are vs. what the model is
- what is the coverage criterion for verification properties?
- w/o an executable specification, the connection to the implementation isn't clear
- real-time behavior isn't handled well
- verification can't handle everything
- having to re-model for verification (especially manually) creates another disconnect from the design and implementation
- need automatic methods for translation between representations
- would like to do the verification as far "up front" as possible—verify design models rather than the source code
- need qualified code generators (but will continue to have to test, just as is done with compilers)
- systems engineering/integration are the tough problems
- using restricted subsets of languages that are certified/verifiable help
- engineering effort required to create specifications to be verified
- formal verification requires an extra level of effort—translating requirements to a formal representation
- need standardization of models to make it worth the effort to use them
- industry lacks knowledge on how to use the tools and advances in verification methods
- need to fix the requirements using executable models—building the model helps people get it right
- need tools for requirements analysis
- requirements evolve, additions keep being made, leading to inconsistencies
- tools that help building requirements—point out errors as they are built—would help
- academic perspective: scalability, lack of background, hybrid systems
- need crisp notions of when tools/methods will or won't work
- we don't know how to do abstraction
- industrial perspective: lack of standards for notations, lack of cost-benefit analysis, lack of training/materials/courses, when does it work?, regulatory hurdles, integration with existing processes, legacy code/engineers
- lack of qualified tools (avionics perspective)

- acceptance takes a long time—10 year process
- implicit specifications can be used more easily (e.g., run-time specifications) because they don't require engineering
- we should have focused/modest ambitions
- having models from MBD has changed things
- engineering process barriers—where the workload is and who gets credit for it
- need management buy in
- need more communication/exposure at the management level
- what are the cost-benefit metrics?
- licensing obstacles
- would like open source/free
- evaluation periods are too short
- compositional verification is needed
- floating pt/nonlinear arithmetic is needed
- saving money/time alone isn't the only metric—MBD is expensive. There are things that are done better
- are the simulation models good enough for analysis?