For a simulationist, accuracy is paramount. If simulation prediction errors exceed bounds of acceptability defined by requirements setters, then performance enhancements and other niceties are for naught. On the other hand, what good is a simulation that requires far too much time to produce its accurate results? So we pursue time saving enhancements, and niceties.

At the design and implementation end of the spectrum, many see potential payoff in designing for reuse in multi-simulation (“federations”) environments. One can reap huge time savings by lashing existing simulations into a federation of simulations comprising a system level solution. Such reuse requires that the component simulations be interoperable. Interoperability and reuse have been the mantra of the DoD simulation community for over a decade. Results have been mixed, as practitioners learned that communication compatibility is not equivalent to semantic compatibility, and now face the tough issues associated with semantic compatibility.

Even with a monumental second effort, culminating in the DoD High Level Architecture for Modeling and Simulation (HLA) four years ago, a recent survey conducted by the Defense Modeling and Simulation Office found two issues standing alone at the top of a community grand challenges introspection, one of them interoperability. Even with a full frontal assault, the DoD M&S community has not solved the problem satisfactorily. So those who explore reuse for architectural studies will likely find themselves standing on others' cupped hands, rather than on their shoulders.

Another technology offering hope is mixed mode simulation. It has the benefit of offering savings both during design and implementation (through reuse) and during simulation execution (exploiting a tendency to higher efficiency in models of higher abstraction). The idea is to reap the performance benefits of the faster abstract model while keeping it within acceptable accuracy bounds through the strategic deployment of a highly detailed model. If the abstract model and the detailed model have some overlap in the phenomena they simulate, then the multi-resolution modeling problem (#1 on the DMSO list) arises. Even without overlap of simulated phenomena, there is often overlap in context, with differing abstraction levels creating potential inconsistencies.
Multi-Resolution Modeling, and the related Variable Resolution Modeling, have been investigated by the simulation community for over a decade. Paul Davis of Rand has claimed for years that variable resolution modeling is at least a very hard problem. I have explored multiresolution modeling with the goal of ensuring consistency between the varying levels of abstraction. We're pleased with our results but admit they won't be used widely because the technology requires too much effort on the part of the mixed mode simulation designer. Our current approach is to apply space-time constraint optimization techniques, as used successfully in the graphics animation community, to ensure compatibility between a requirements setter's constraints and the behavior of the mixed mode model. Independent of anyone's favorite approach to solving the problem, we should note its position at the top of the DoD simulation community's challenges list (by a wide margin) and recognize there are no quick or certain answers.

As an aside, there has been a persistent movement to address MRM using a technology known as "aggregation-disaggregation," in which a simulation dynamically switches between an abstract mode and a highly detailed mode, as needed. We have documented all sorts of anomalies that can arise using this approach. It is to be avoided.

At the execution end of the spectrum, there lies hope in multi-processor simulations: go after 1000-fold performance gains with commodity processors and networks. Parallel simulation, as the field has been aptly named, has sustained an active research community since 1977. You can find development and hosting environments (e.g. UCLA Parsec, GaTech GTTW), simulation time management algorithms (e.g. Chandy-Misra Null Messages, Time Warp, Elastic Time), and data distribution management algorithms. Despite all of this effort, no one can tell you (correctly) that there is one particular approach that will work well for your application without possessing a complete understanding of your application and its associated inputs. There is simply no general purpose solution. A number of us have shown over the years that within a selected pair of time management algorithms each can perform arbitrarily better than the other under different circumstances. There have been some outstanding results in parallel simulation, with true linear speed-up, but they have tended to be point solutions, yielding little insight for someone else's problem.

So what? Am I suggesting there's no hope with respect to reuse, mixed-mode, or multi-processor simulations? Definitely not. There's been a significant share of disappointments and hard knocks, but on the bright side there are many useful tools and concepts, a lot’s been learned, and there are plenty of promising successes. It's not likely that research into simulation technology for processor/system architectures will identify much low hanging fruit. The simulation community has solved the easier problems for most everybody. The best opportunities reside in a small set of hard issues that need to be addressed to completion. As the architecture community addresses questions about reuse, abstraction, mixed modes and sensitivity, it will find many simulationists with a wealth of experience on those topics. Let’s get started.