

***Developing and Analyzing Architectures
for Large Real Time Systems
from UML Diagrams***

**Presented at: MDA Implementers' Workshop:
Succeeding With Model Driven Systems**

**by: Leonard Weinberg, Rob Cloutier, and Andrew Winkler
Lockheed Martin Maritime Systems & Sensors**

on: December 5, 2003

Introduction

- Lockheed Martin has more than 30 years experience in designing and building computing systems for US Navy cruisers and destroyers
- Systems are large and demanding (12,000,000 SLOC in >50 computers)
 - many use real time O/S
 - computer utilization >50 %;
 - message latencies in the milliseconds
 - automatic reconfiguration within seconds of failure
- Currently, requirements are in paper form, and computer program development is people-intensive
- For next generation systems, we have begun development of the architectures using UML to analyze and document the systems
- Over the last seven years, event driven computing system architecture models have helped shape the computer program designs and to predict and map their performance on target systems
- MDA must include a [capability for modeling computing system architectures \(CSA\)](#)

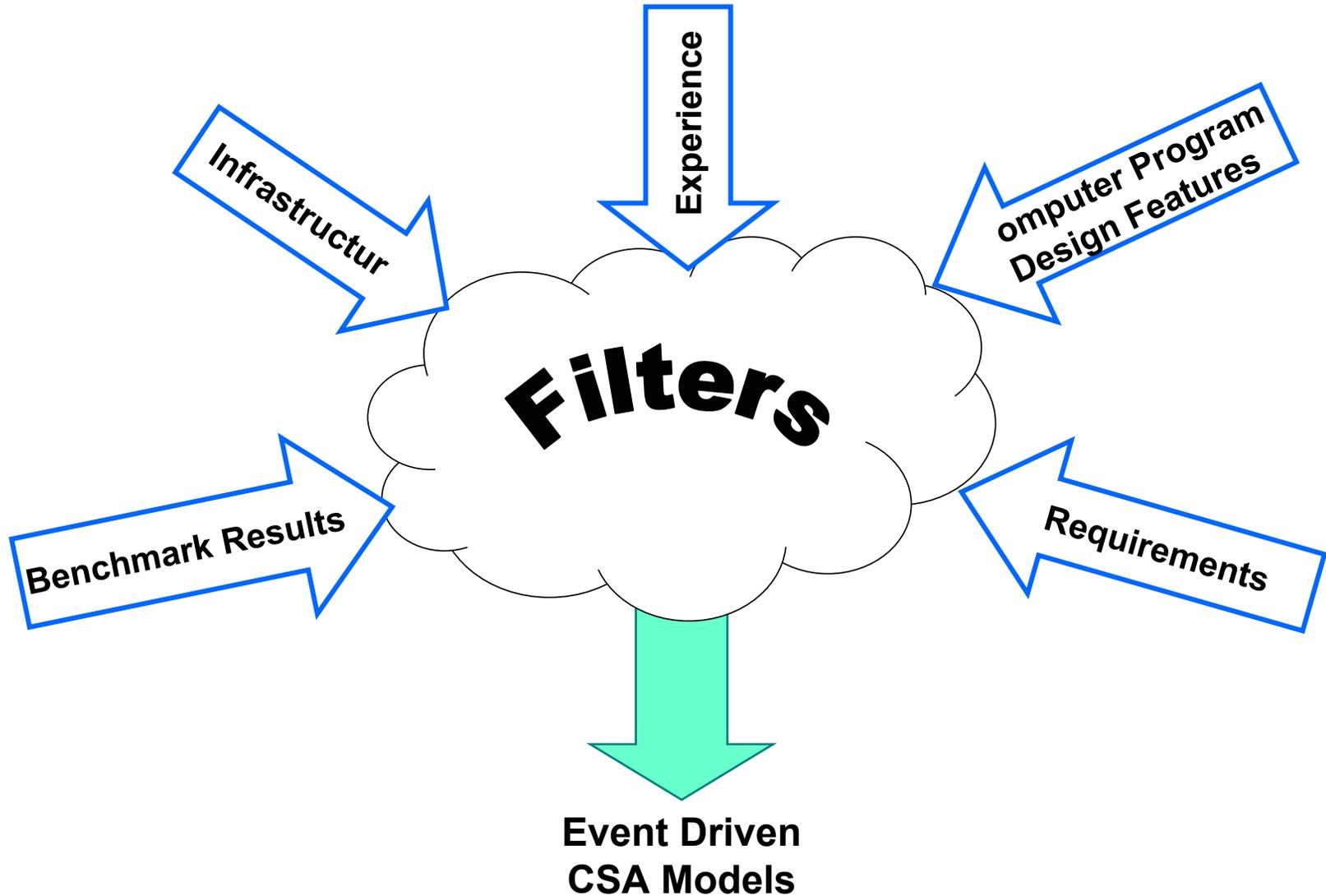
Elements of a Computing System Architecture (CSA)

- **Infrastructure**
 - **Computers:** including their input-output devices, clocks, etc
 - **Connectivity:** fibers or cabling, and the network switches
 - **Support software:** operating systems, and other support programs
 - **Special equipment:** data converters, data recording devices, etc

- **Application software**

- **Computer program design and performance, computer program allocations, scheduling, reconfiguration, utilization, latencies, etc,**

Current Approach for Developing our Event Driven CSA Models

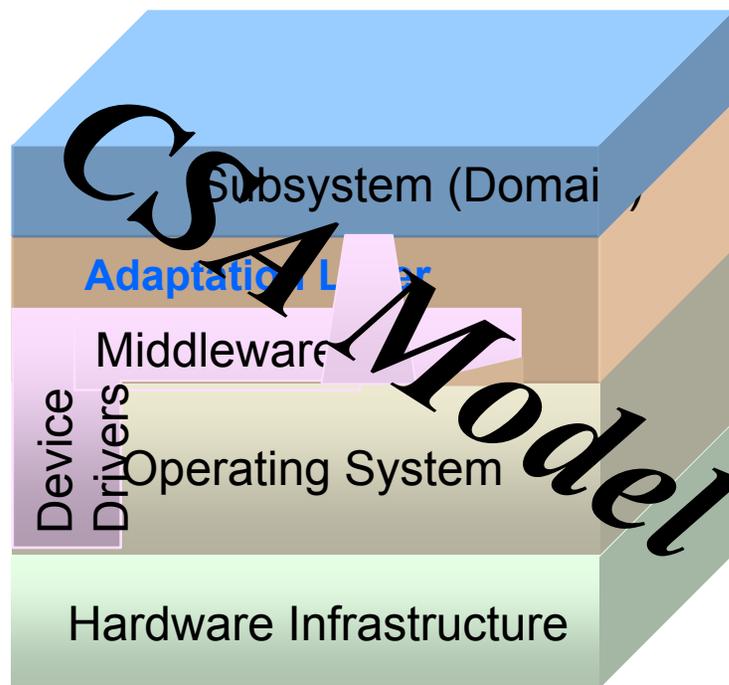


Computing System Architecture Modeling

Inputs and Outputs

Modeling Inputs

- System requirements, budgets, goals
- Infrastructure design, requirements, benchmarks
- Computer program design, characteristics, allocation to computers, service times
- Communications traffic-average & worst case scenarios



Modeling Outputs

Estimated Performance

- How well requirements are met, how computer and network are utilized, how consistently critical message latencies are satisfied, whether transfer queues ever fill up

Predicted Risk

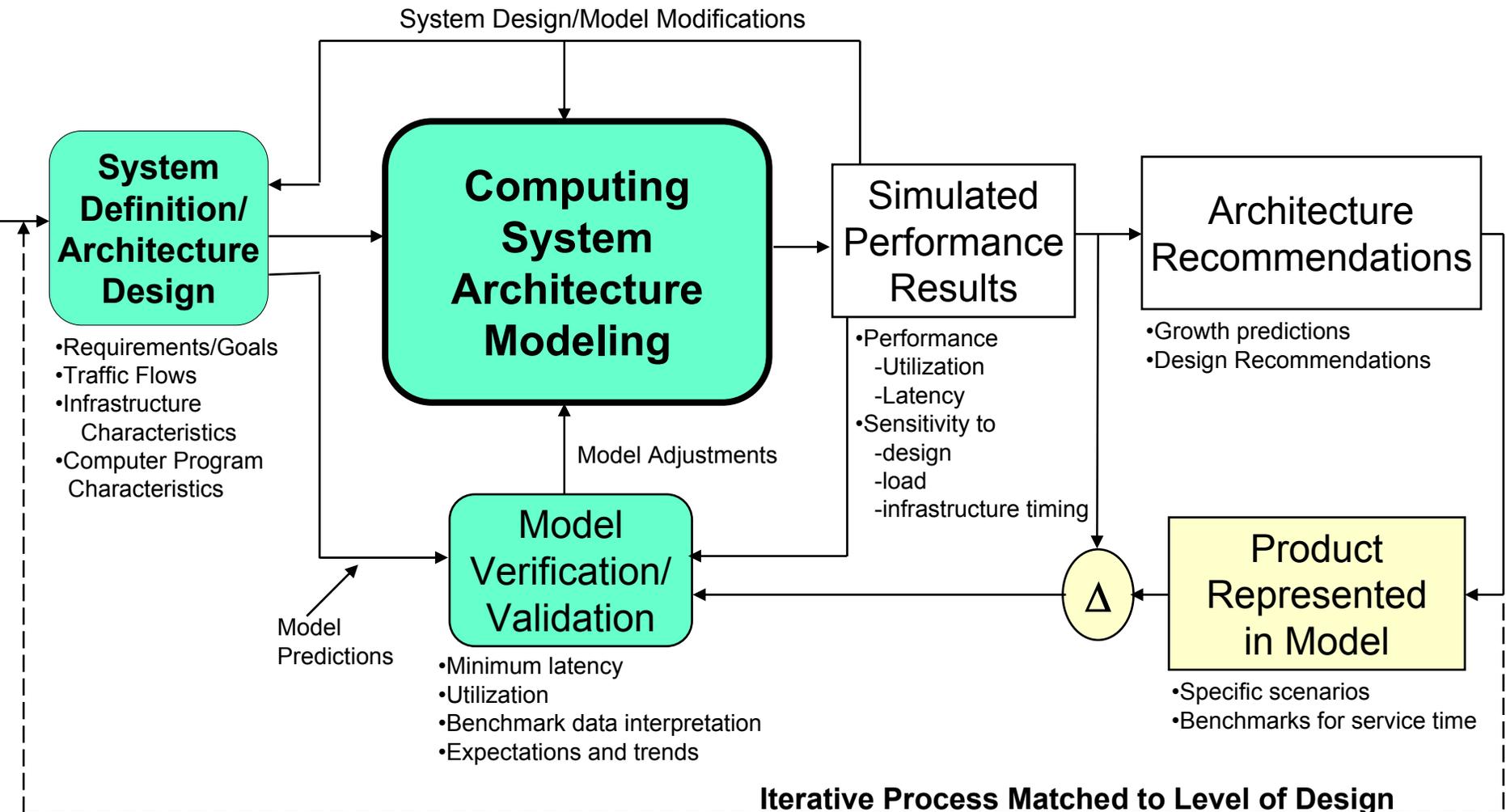
- Current design
- Growth potential

Recommendations

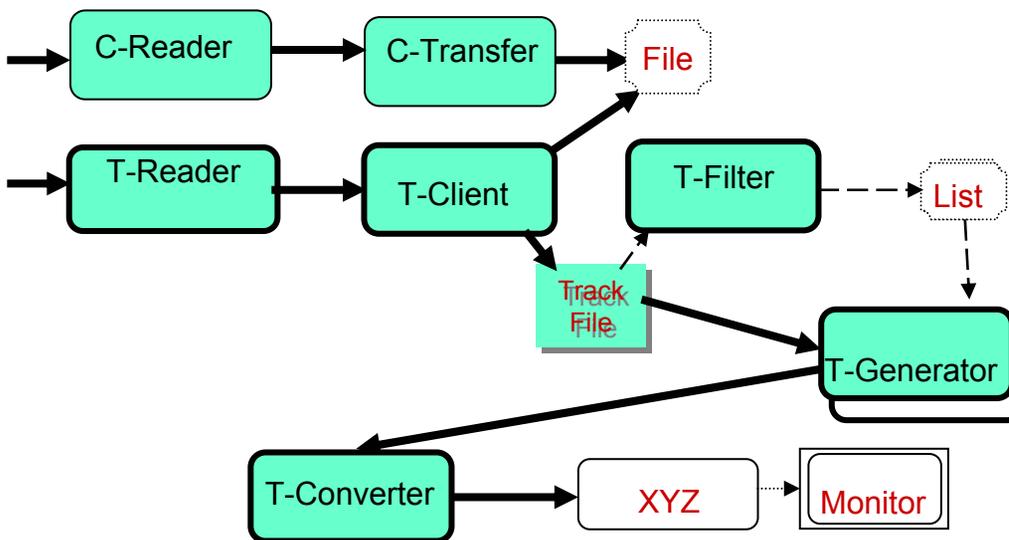
- Modifications to system budgets, to equipment and software design, to allocations and algorithms,

*The Performance Model Provides a Virtual Computing System Architecture
The Model May be Built in Pieces or in Total and Maintained in Spreadsheets*

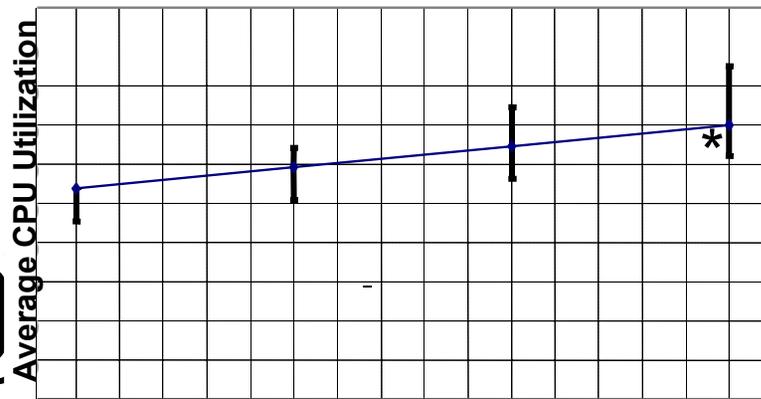
Current Computing System Architecture Analysis & Simulation Process



Steps in the CSA Modeling Process

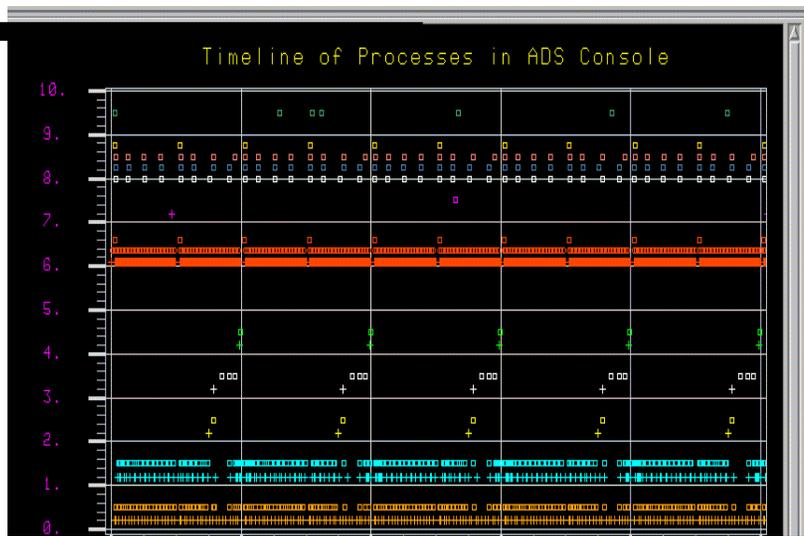


Program Design Flow, Scheduling & Timing Information



Track Rate (tracks per second)

Performance Results



Simulated Process Timeline



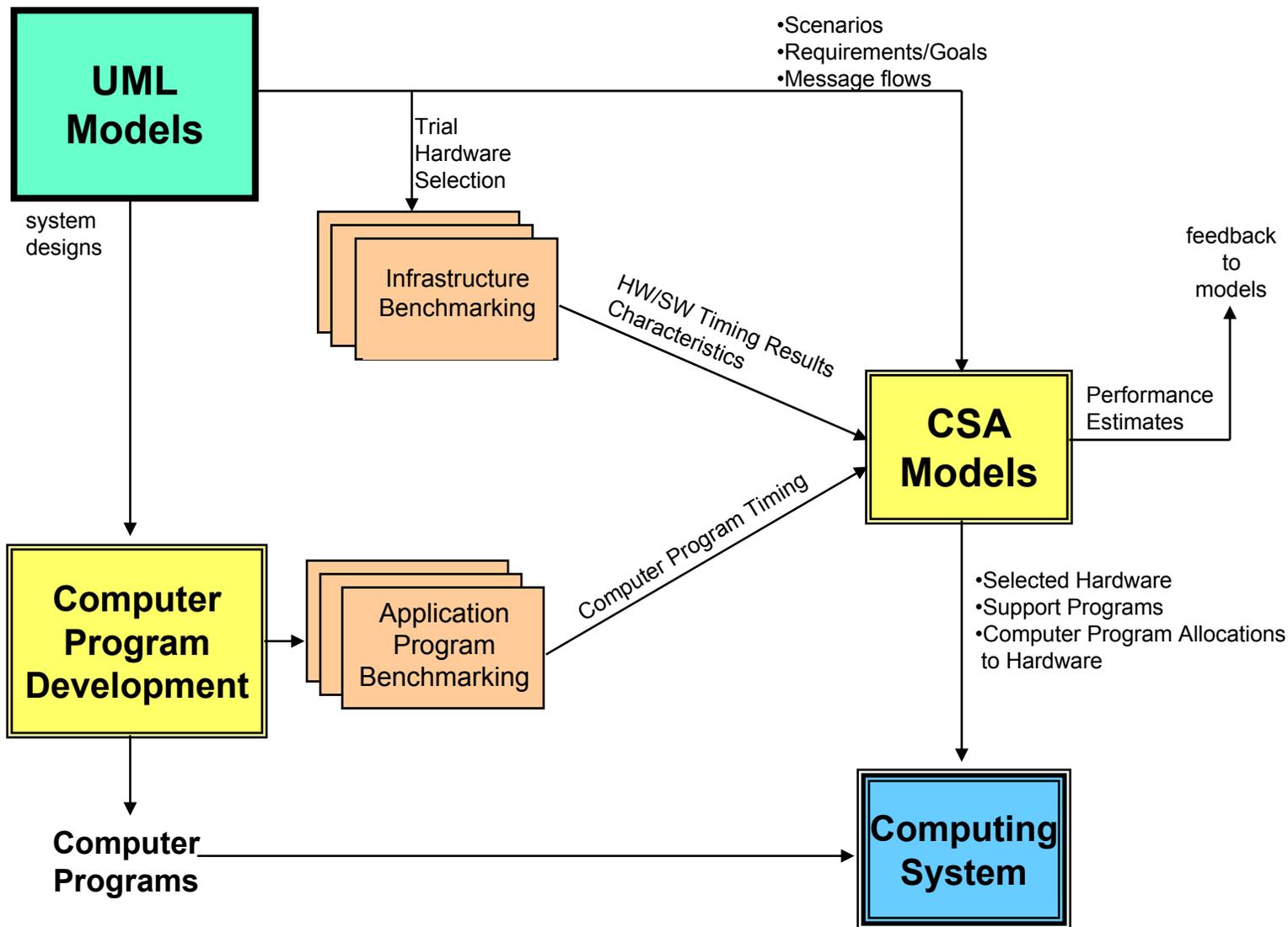
Instability Exhibited During Simulation

Objectives in Transitioning to MDA Design

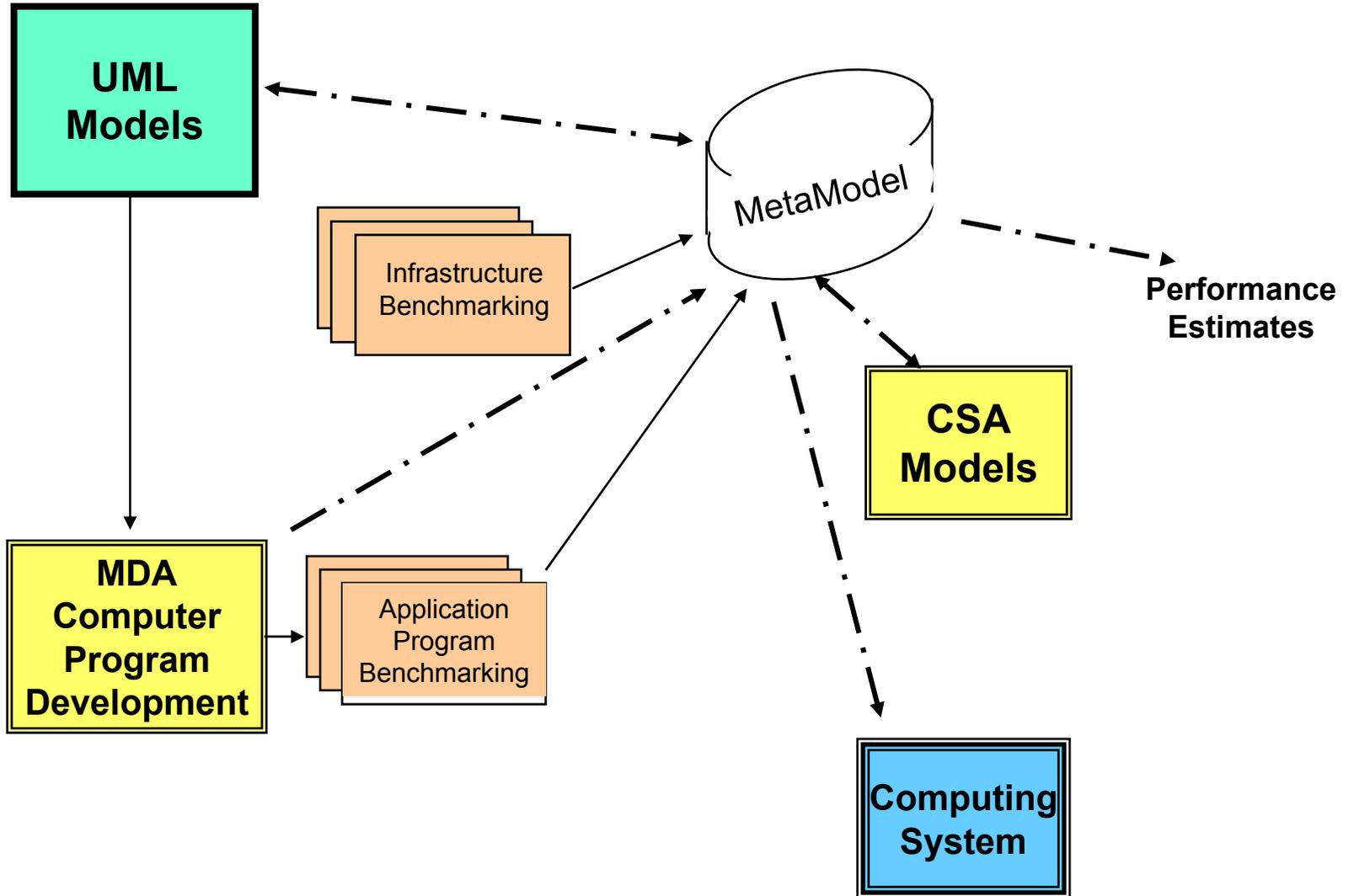
- **Computing System Architecture (CSA) modeling must become an automated process with interfaces to the UML modeling tool**
 - **Results from CSA models will affect the UML models themselves, and the final architectural design decisions**

- **CSA in an MDA environment requires a common data repository and system of analyses to accept inputs from the UML models and to extend them to the larger system**
 - **With an investment of effort and imagination, this repository may grow into a system metamodel**

MDA Using CSA & Code Development Models



Potential Future Meta-Model Implementation



Computing System Architecture Modeling Tool Requirements

- **Must easily reallocate SW to HW independently throughout the Computing System Architecture**
 - **Software allocations to computers are dynamic throughout the development process and beyond. Independent control of the SW and HW is very important. Not all event driven tools provide for this independence.**
- **Must create components for high level abstraction and implementation speed**
- **Must interface with UML model data**
- **Must easily save and recall model runs and configurations**

Summary

- **As we move to model-driven architecture development, the CSA approach will support the system design, and help to estimate system performance**
- **By implementing a metamodel to facilitate computer system architecture modeling, a higher degree of automation and consistency in the computing system architecture should result**