Computing Architecture Performance Prediction and UML Requirements

Presented to MAST 2006 on September 5, 2006 by
Leonard Weinberg
Charlie Hobbs

Introduction and Summary

Lockheed Martin has more than 30 years experience in designing and building computing systems for U.S. Navy cruisers and destroyers

Systems are large and demanding (12,000,000 SLOC in >50 computers)

- Many use real-time O/S
- Message latencies in the milliseconds
- Automatic reconfiguration within seconds of failure

Computer program development is now using distributed component software architectures using UML model driven techniques to document requirements and designs

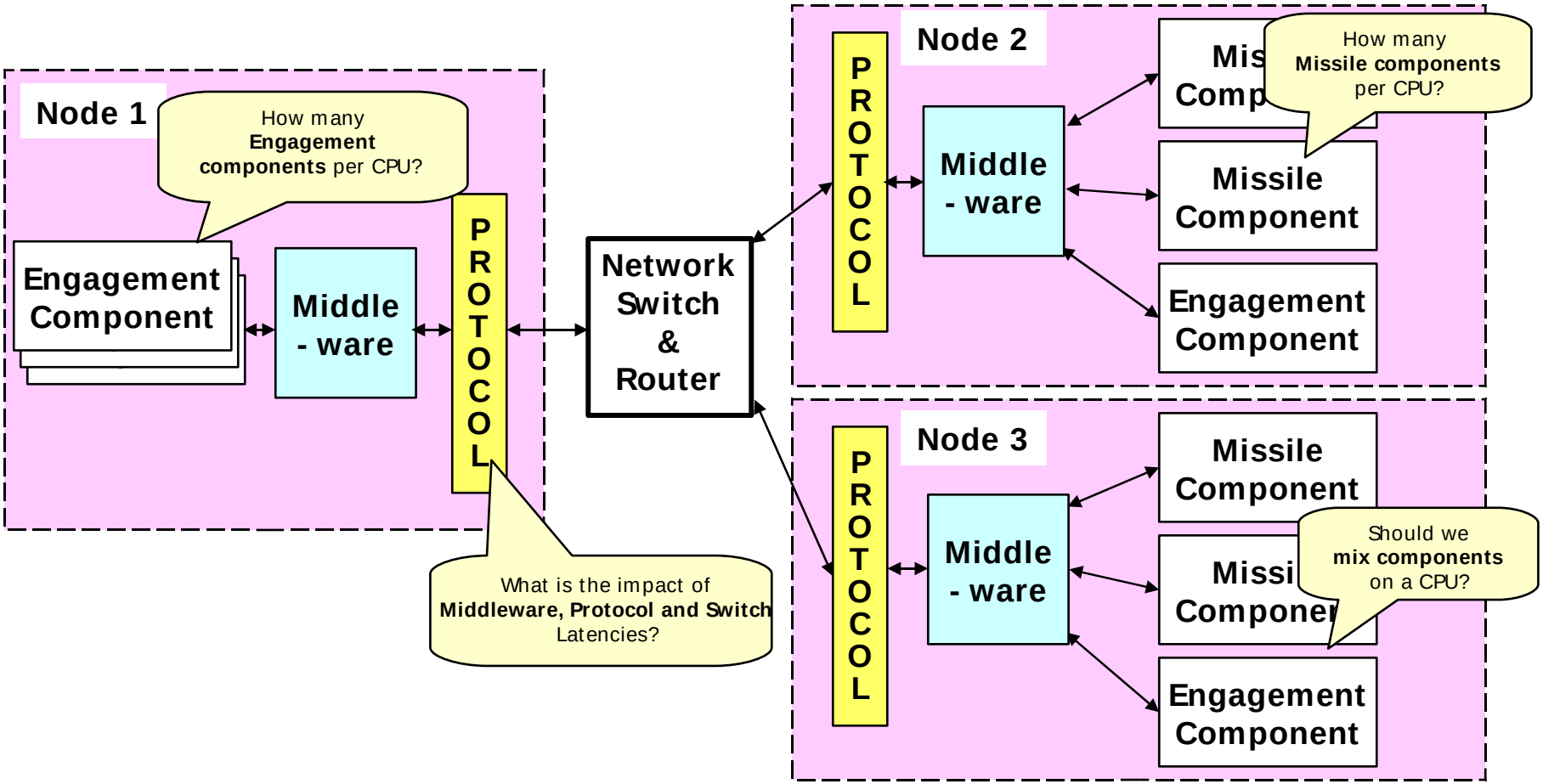
Tasked to develop an easy to use system engineering product to support investigations of proposed UML designs to:

- evaluate, predict & optimize performance
- provide early assessment of system designs
- assess impact of future war-fighting capability

Algorithms, spreadsheets and existing discrete event simulation tools can not easily capture the dynamic and statistical nature of the computing environment

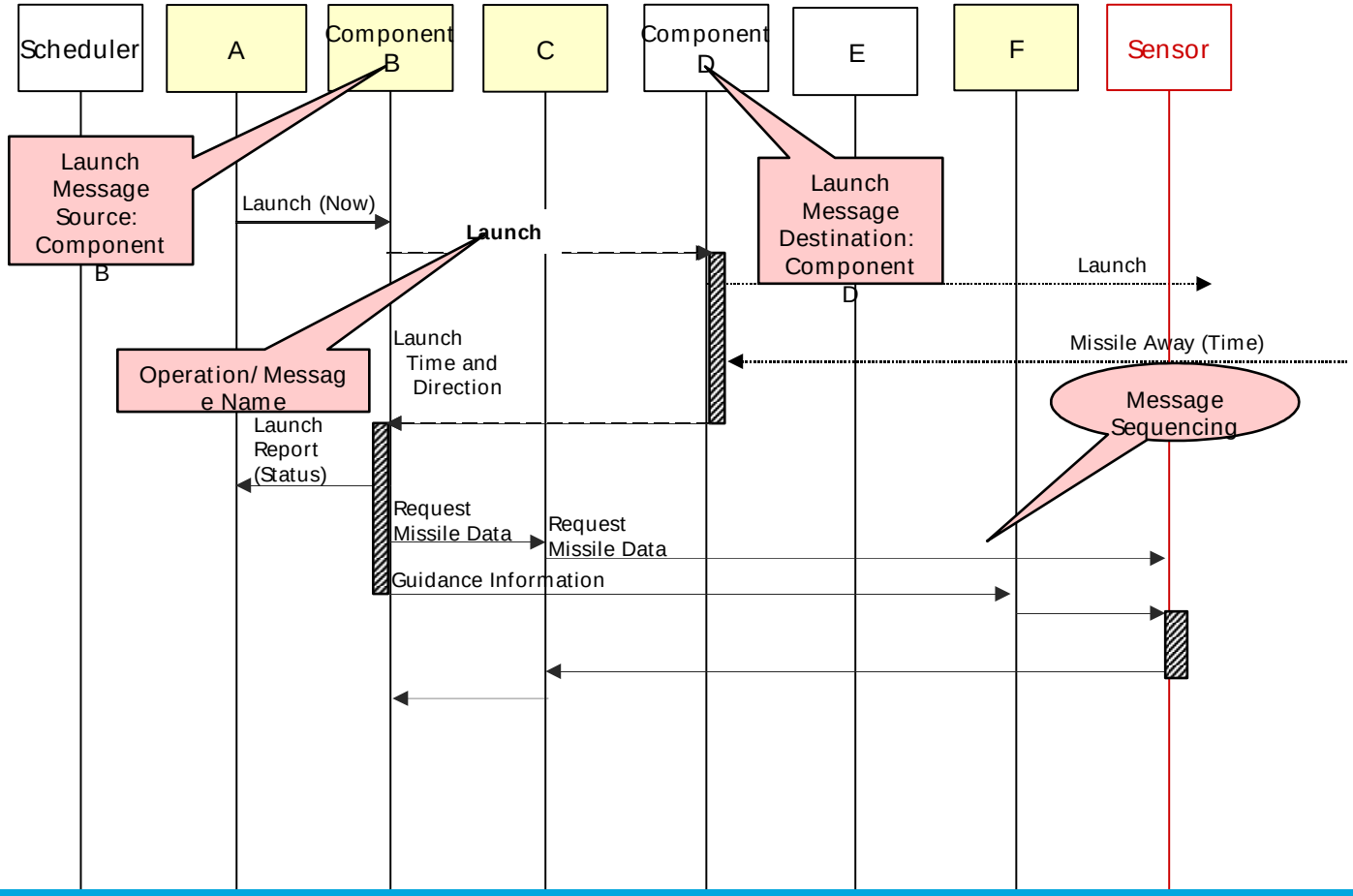
Infrastructure Modeling Assemblies **have been created to enhance an existing LM event driven simulation tool, CSIM, to speed, simplify & mimic UML Sequence Diagram Operations**

Computing Architecture Development Begins with a Qualitative Vision for a Distributed Computing Architecture



Guidance Loop Closure Times Drive Component to Node Allocation

Software Requirements are Expressed by UML Models. UML Sequence Diagrams Express Message Flows.



How do we rapidly Capture Message Sequencing and Component Deployment in an Event Driven Simulation Tool?

Building a Dynamic Performance Analysis Environment

Our Solution

Using the Lockheed Martin discrete event simulation tool, CSIM, we have built Infrastructure Modeling Assemblies

The IMA is a model of a reasonably large infrastructure assembly, representing the processing flow initiated by the transmission of a single message in a computer

- It may include processing by an application or middleware, and be governed by priority, scheduling rules, etc.
- Each IMA is "tuned" by entering values for architectural details

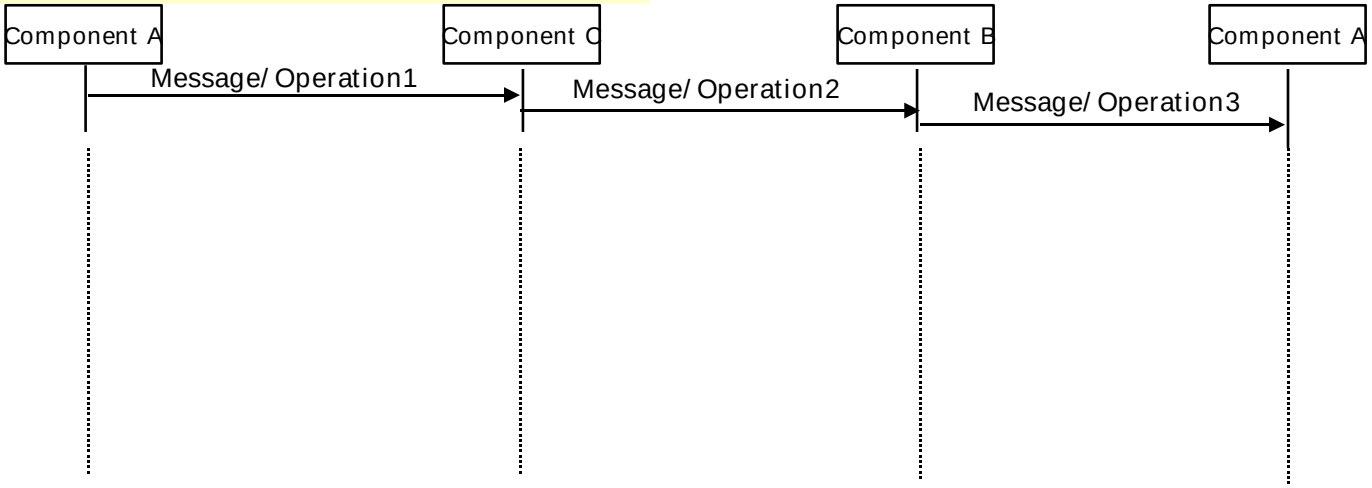
IMAs simplify building the performance model

- By connecting IMAs, we emulate a Sequence Diagram of any complexity

Experience indicates the large savings possible by modeling with and re- using Infrastructure Modeling Assemblies

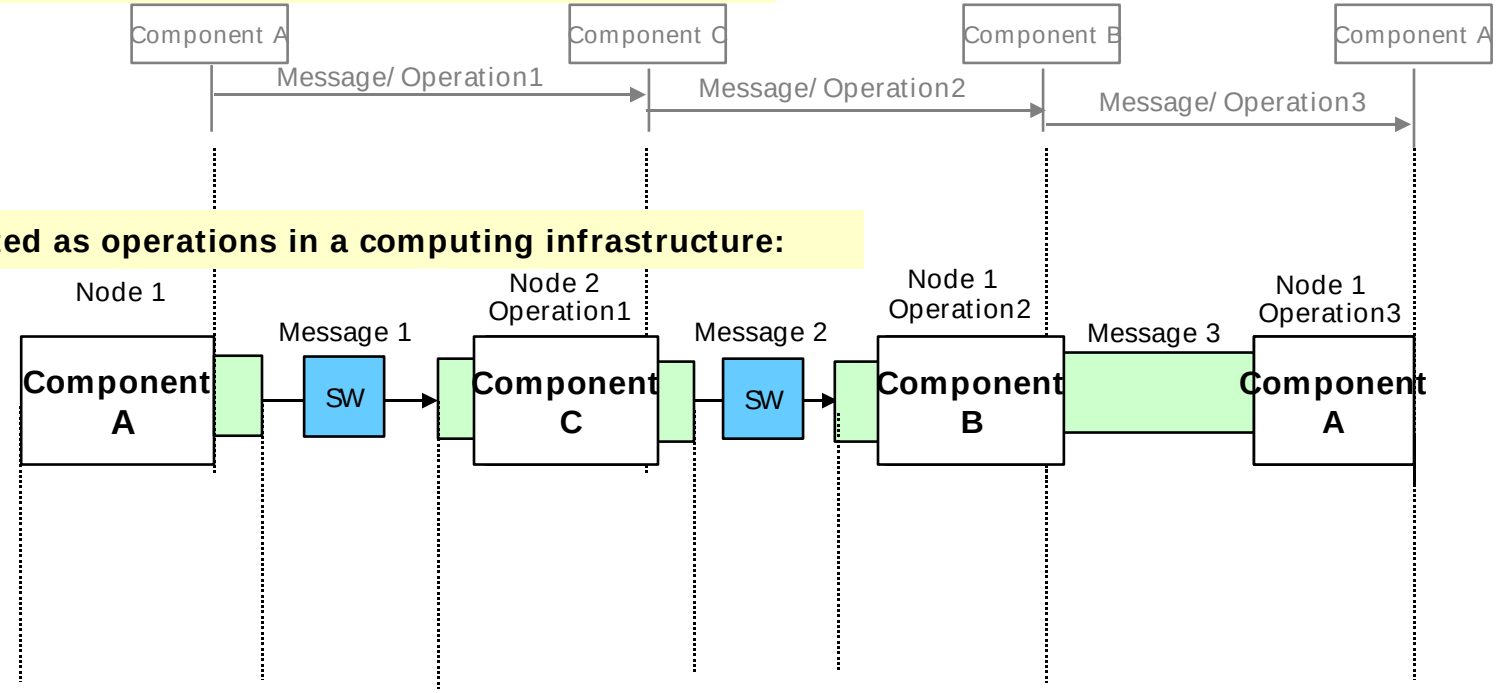
Sequence Diagrams can be Viewed in Terms of Infrastructure Modeling Assemblies- 1

- **Message flows & operations in a sequence diagram:**



Sequence Diagrams can be Viewed in Terms of Infrastructure Modeling Assemblies- 2

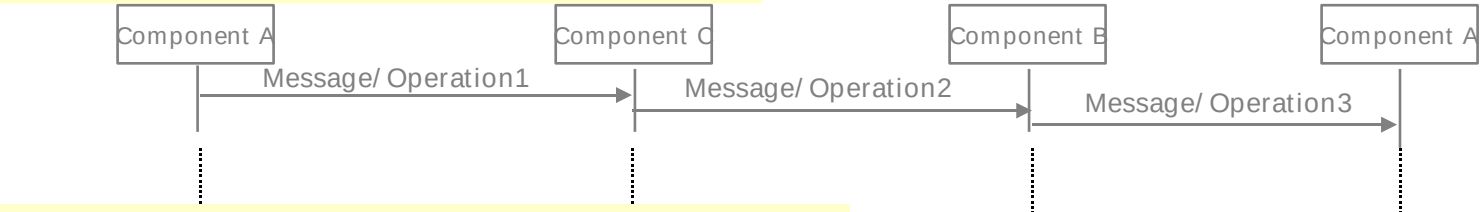
- **Message flows & operations in a sequence diagram:**



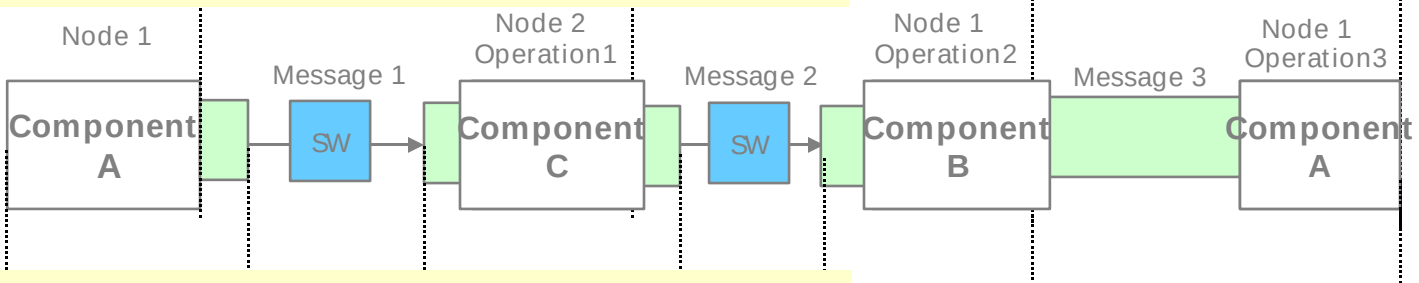
- **interpreted as operations in a computing infrastructure:**

Sequence Diagrams can be Viewed in Terms of Infrastructure Modeling Assemblies- 3

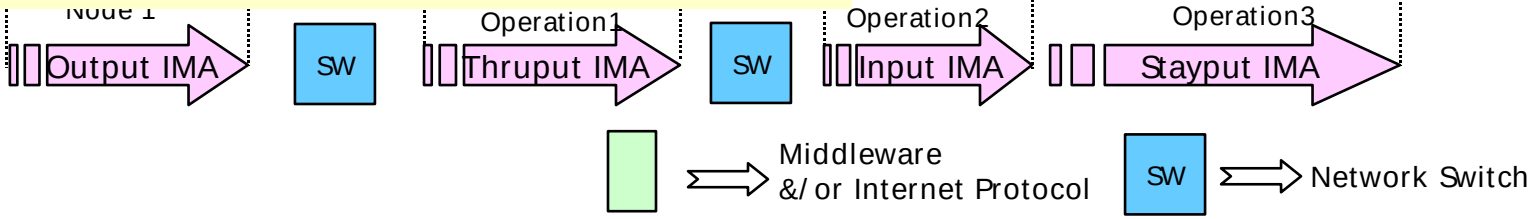
- **Message flows & operations in a sequence diagram:**



- **interpreted as operations in a computing infrastructure:**

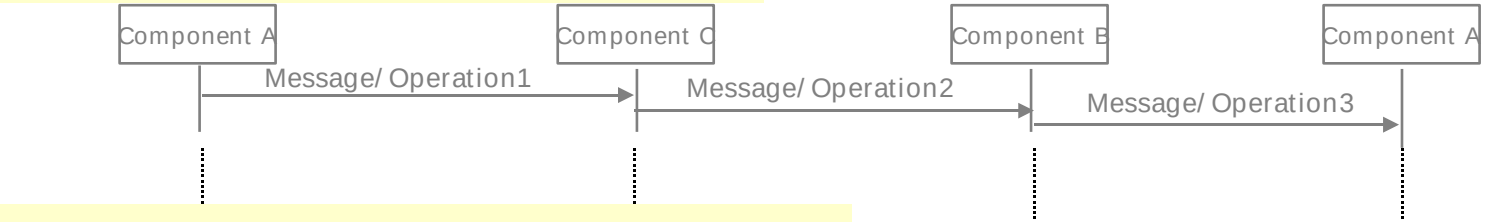


- **or message flows through infrastructure modeling assemblies:**

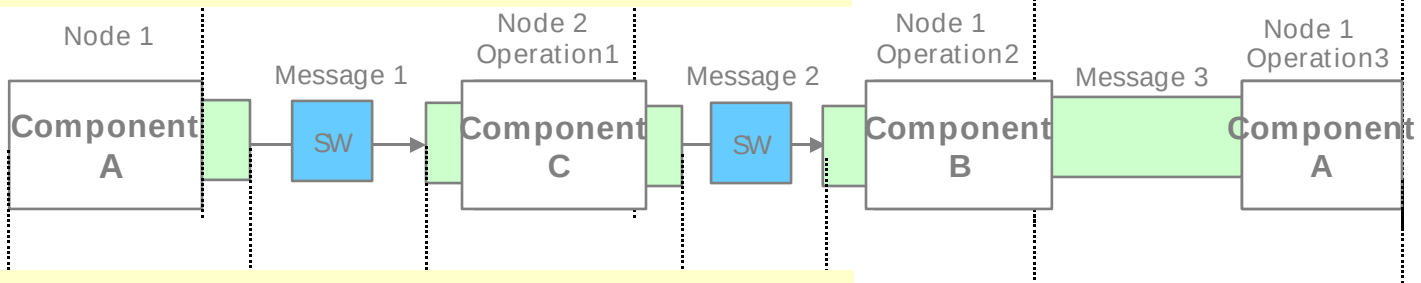


Sequence Diagrams can be Viewed in Terms of Infrastructure Modeling Assemblies- 3

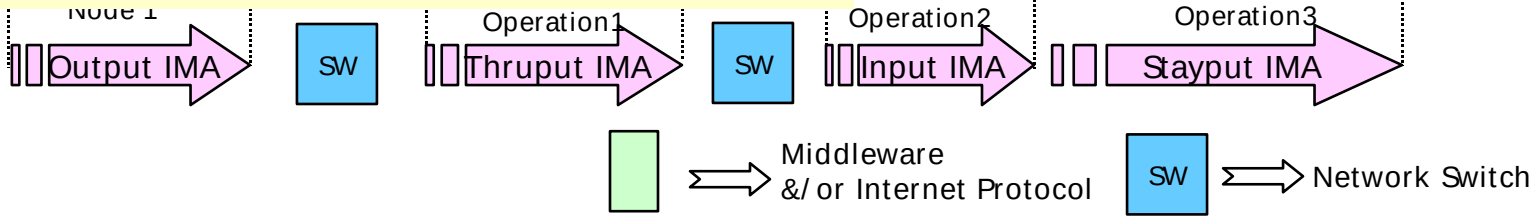
- **Message flows & operations in a sequence diagram:**



- **interpreted as operations in a computing infrastructure:**

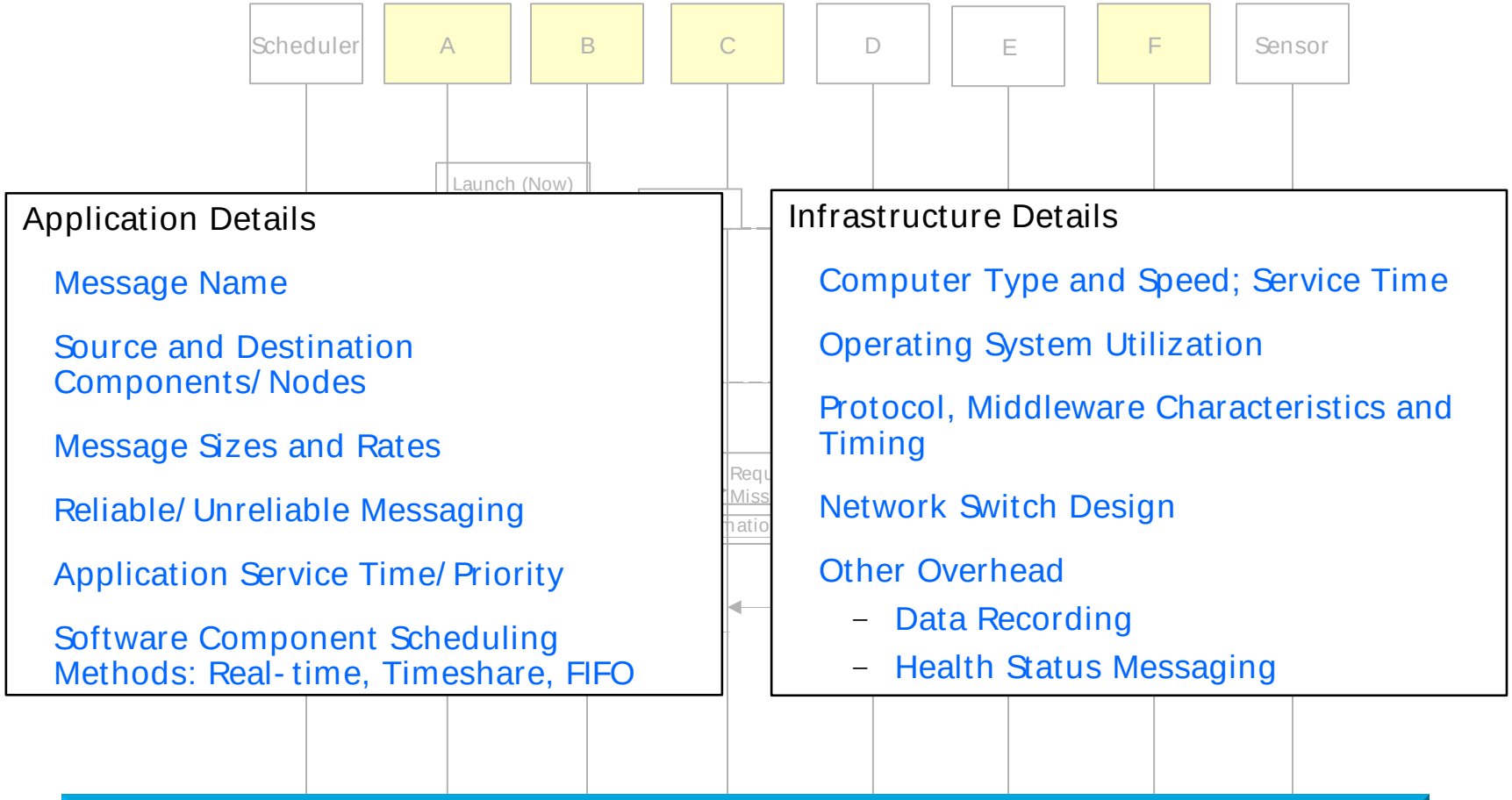


- **or message flows through infrastructure modeling assemblies:**



IMAs Have Been Incorporated into the Lockheed Martin CSIM Event Driven Modeling Tool

Architectural Details Have Been Incorporated into our UML Models to Support Performance Modeling



Initial Engineering Estimates are Fine Tuned as Development Progresses

Lessons Learned Using CSIM Performance Modeling Tool To Predict Weapon Control Distributed Component Computer Program Performance Based on UML Requirements

Achieved ten- to- one speedup in designing performance models

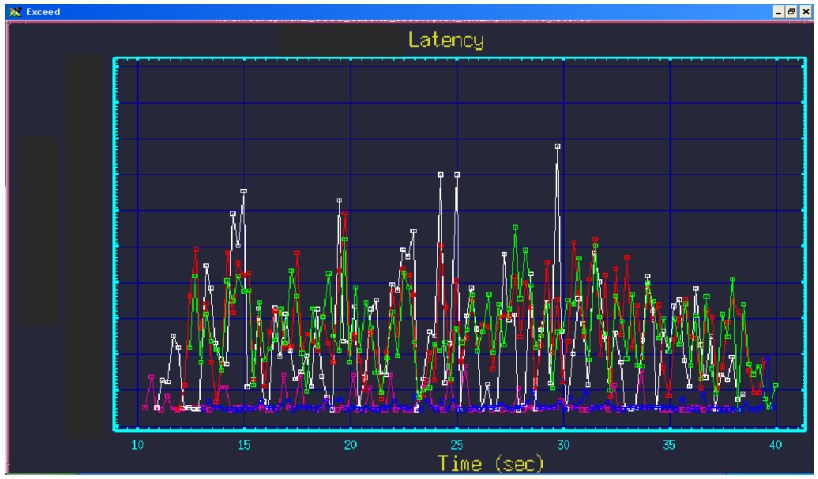
- UML models capture data previously obtained directly from engineers
- IMAs help speed up construction time, with fewer errors

Selecting the correct software design level is critical to building a satisfactory performance model

- If level is too low, building the model becomes complicated
- If level is too high, contention among schedulable components is lost and latency results sacrificed

Even though computer speeds have increased dramatically, CPU utilization levels still must be maintained on the order of 50%, for this type of application

Mixing components of different types on a single node tends to reduce spikes in latency



From UML Requirements to Computing Architecture Performance Prediction