Validation of Safety-Critical Systems with AADL

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Peter H Feiler

April 11, 2008

Outline

Multiple aspects of system validation

System & software engineers working together

Multi-fidelity model-based analysis

Property preserving transformations

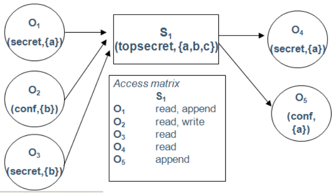
Conclusions

Dimensions of System Validation

Validation of models against system

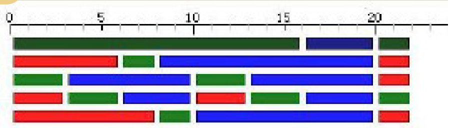
The system

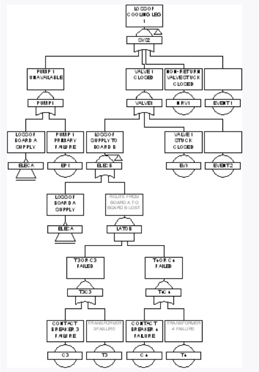
Model-based vof system



Access matrix

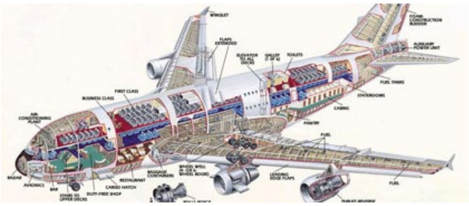
	S_1
O_1	read, append
O_2	read, write
O_3	read
O_4	read
O_5	append





System models

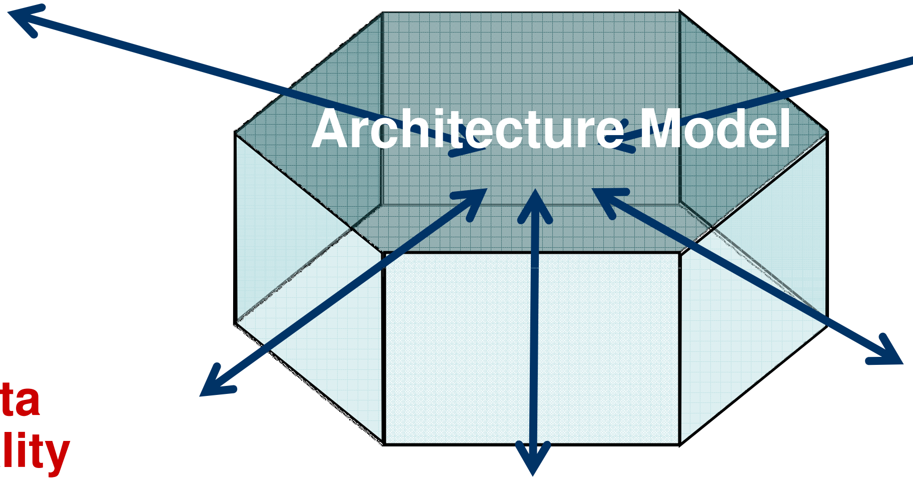
Validation of implementation against system models



System implementation

Single Source Annotated Architecture Model

Predictive Analysis Across Engineering Dimensions



Architecture Model

Availability & Reliability

MTBF

FMEA

Hazard analysis

Security

Intrusion

Integrity

Confidentiality

Data Quality

Data precision/ accuracy

Temporal correctness

Confid

Real-time Performance

Execution time/ Deadline

Deadlock/starvation

Latency

Resource Consumption

Bandwidth

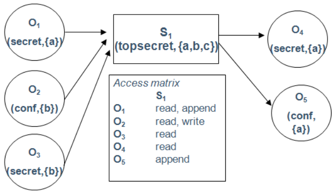
CPU time

Power consumption

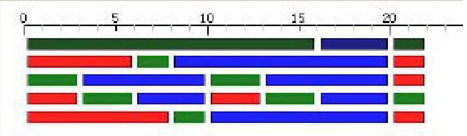
Fewer independently developed models reduces model validation

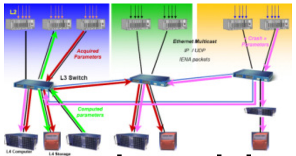
Low incremental cost for additional analyses & simulations

Architecture-Driven Modeling

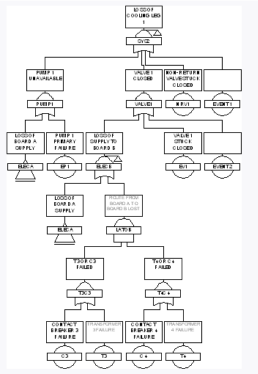


Automatically derived analytical models

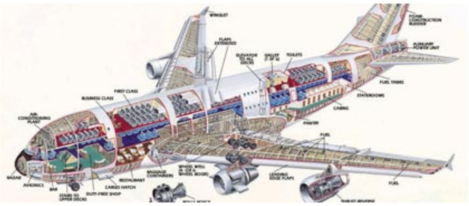




Annotated architecture



System generation from validated models



Validation of generators

AADL and Safety-Criticality

Fault management

- Architecture patterns in AADL
  - Redundancy, health monitoring, …
- Fault tolerant configurations & modes

Dependability

- Error Model Annex
- Specification of fault occurrence and fault propagation information
- Use for hazard and fault effect modeling
- Reliability & fault tree analysis

Behavior validation

- Behavior Annex
- Model checking
- Source code validation

Outline

Multiple aspects of system validation

System & software engineers working together

Multi-fidelity model-based analysis

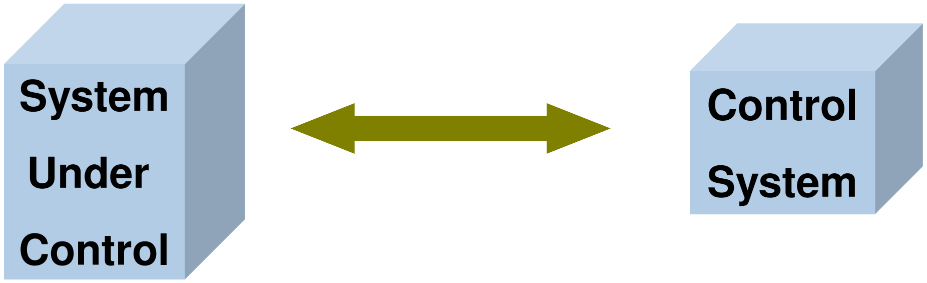
Property preserving transformations

Conclusions

Traditional Embedded System Engineering

System Engineer

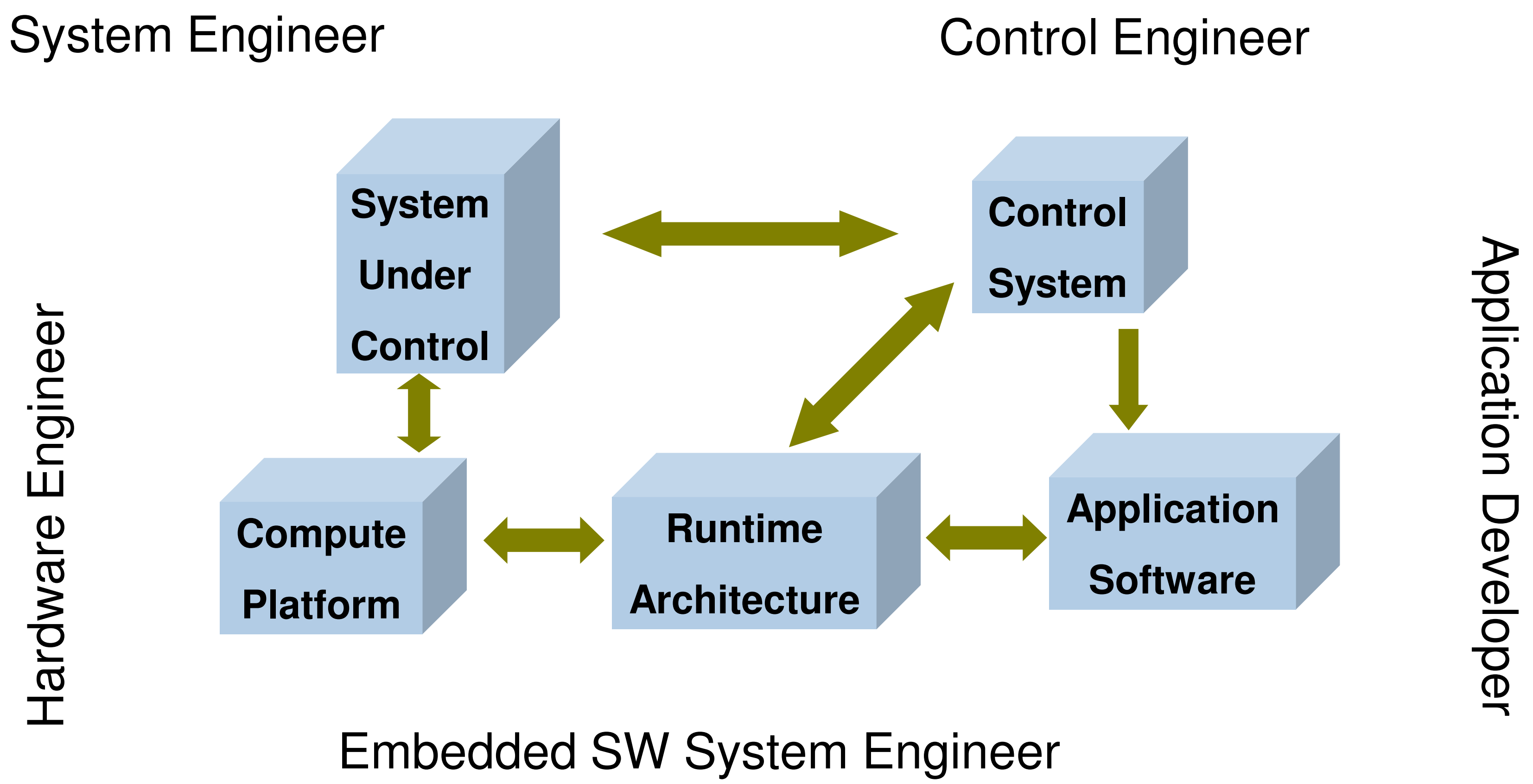
Control Engineer



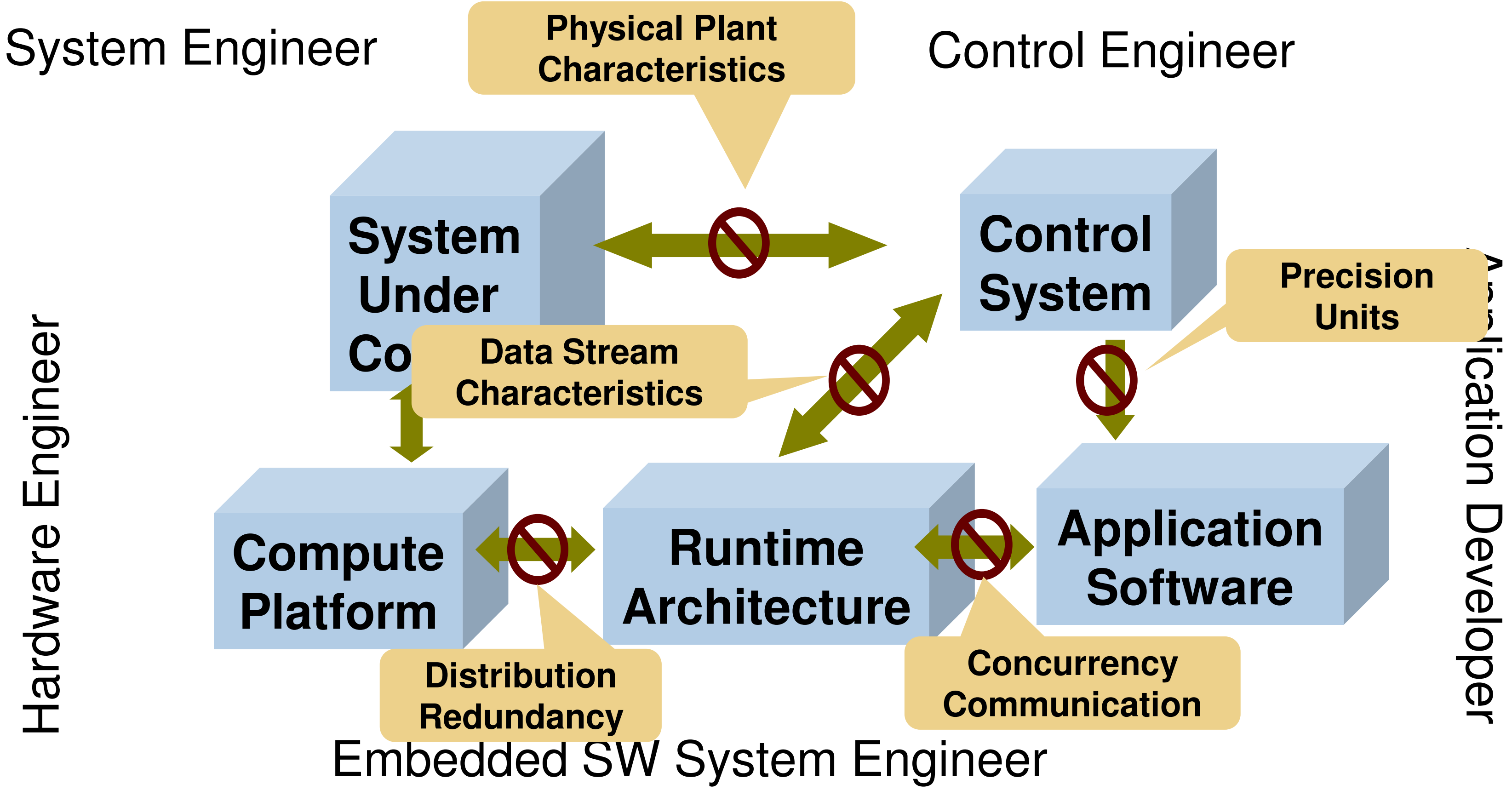
System Under Control

Control System

Software-Intensive Embedded Systems



Mismatched Assumptions



System Engineer

Physical Plant Characteristics

Control Engineer

System Under Control

Control System

Precision Units

Data Stream Characteristics

Hardware Engineer

Application Developer

Compute Platform

Runtime Architecture

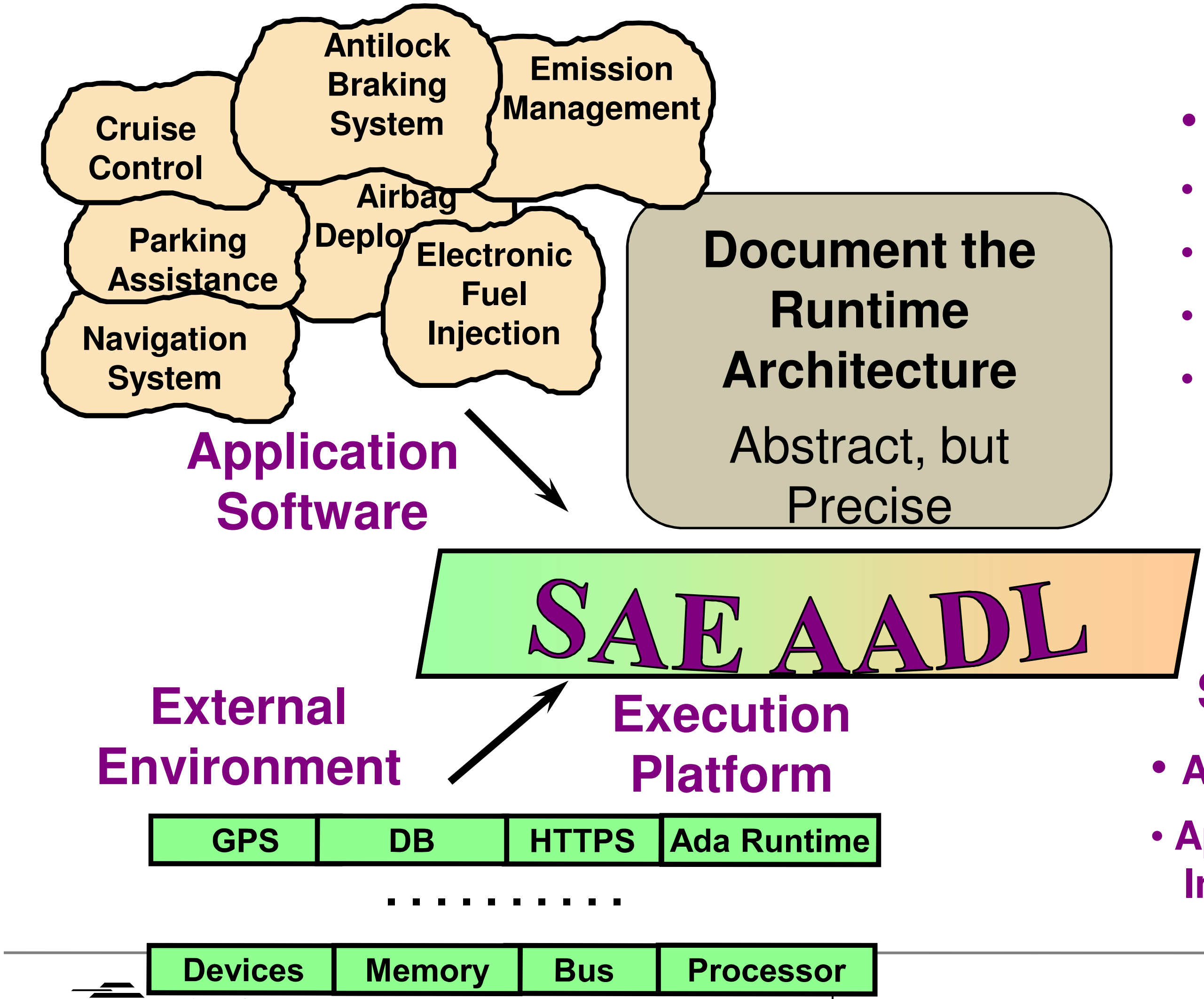
Application Software

Distribution Redundancy

Concurrency Communication

Embedded SW System Engineer

Predictable Embedded System Engineering



Antilock Braking System

Emission Management

Cruise Control

Airbag Deplo

Parking Assistance

Electronic Fuel Injection

Navigation System

Application Software

Document the Runtime Architecture

Abstract, but Precise

SAE AADL

External Environment

Execution Platform

GPS	DB	HTTPS	Ada Runtime

. . . .

Devices	Memory	Bus	Processor

System Analysis

- Schedulability
- Performance
- Reliability
- Fault Tolerance
- Dynamic Configurability

System Construction

- AADL Runtime System
- Application Software Integration

Working Together

Conceptual architecture

- UML-based component model
- Architecture views (DoDAF, IEEE1471)
- Platform independent model (PIM)

System engineering

- SysML as standardized UML profile
- Focus on system architecture and operational environment

Embedded software system engineering

- SAE AADL
- OMG MARTE profile based on AADL
- AADL as MARTE sub-profile
- Non-functional properties require deployment on platform

Data modeling

- UML, ASN,, …

Outline

Multiple aspects of system validation

System & software engineers working together

Multi-fidelity model-based analysis

Property preserving transformations

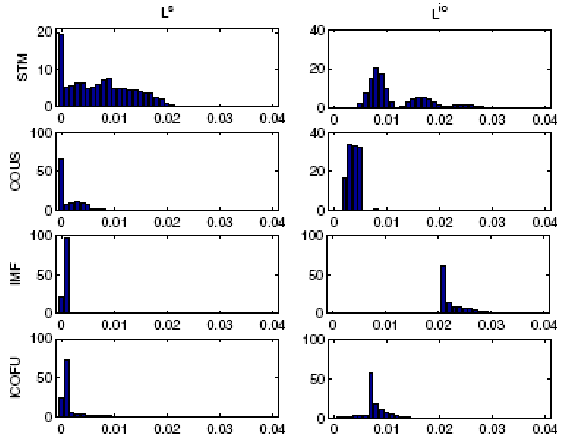
Conclusions

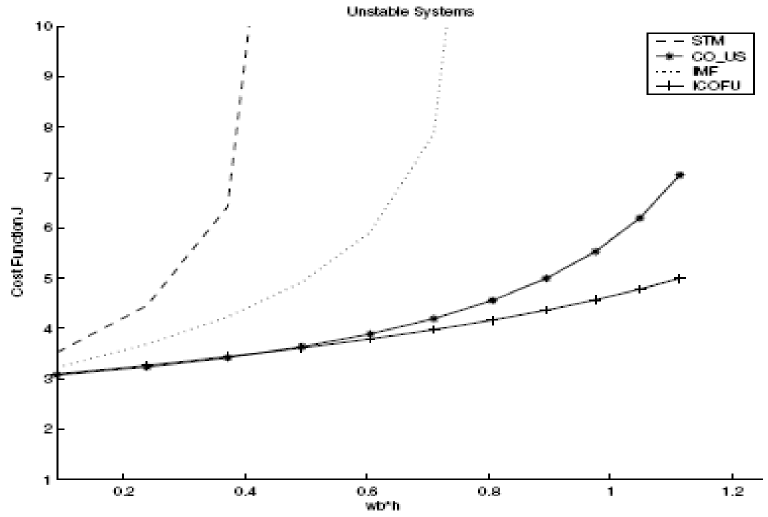
Impact of Sampling Latency Jitter

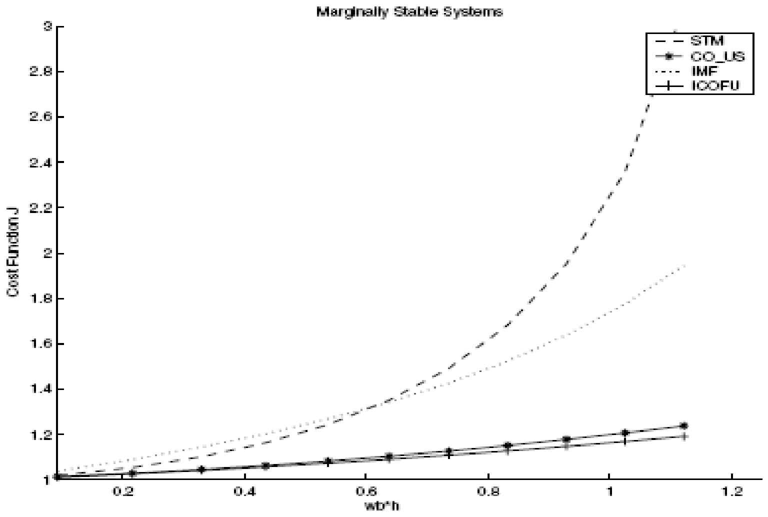
Impact of Scheduler Choice on Controller Stability

- A. Cervin, Lund U., CCACSD 2006

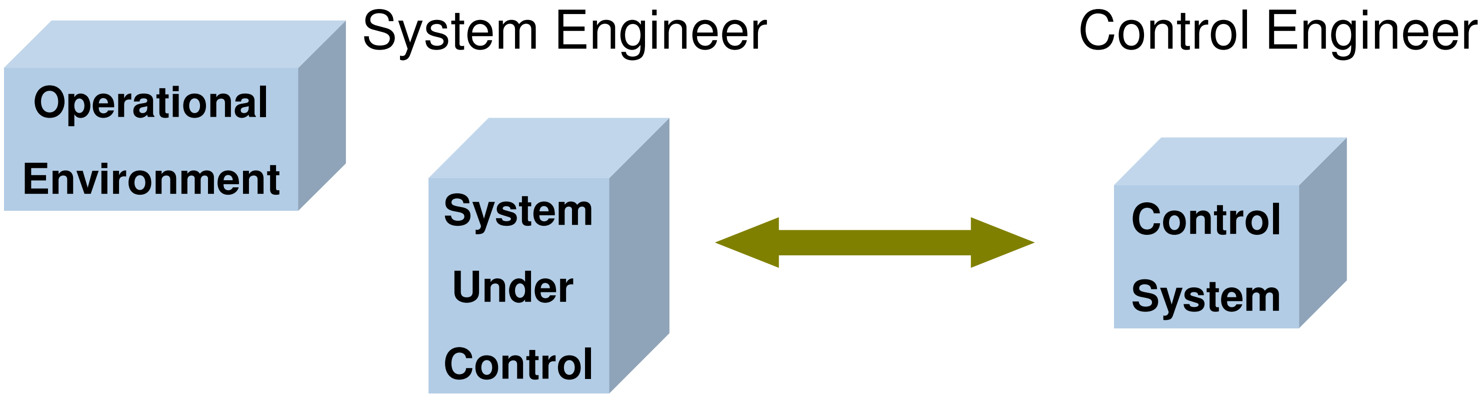
Sampling jitter due execution time jitter and application-driven send/receive







Latency Contributors

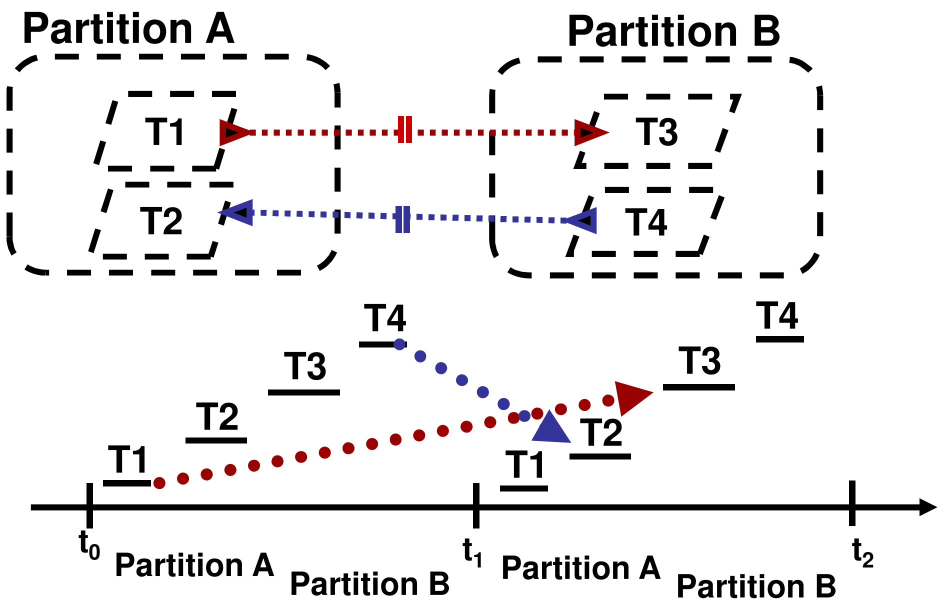


- Processing latency
- Sampling latency
- Physical signal latency

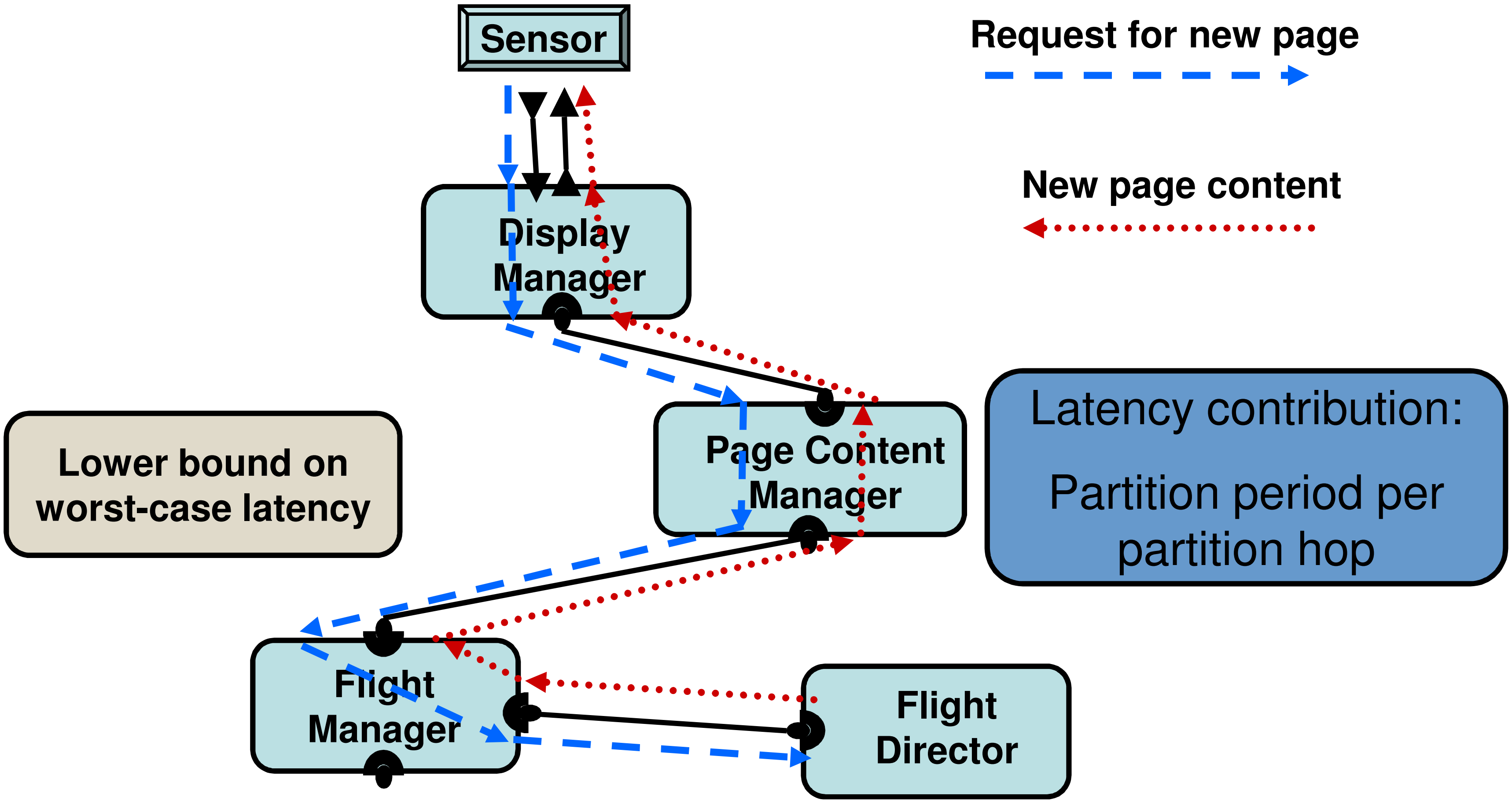
ARINC 653 Partitions & Communication

Frame-delayed inter-partition communication

Timing semantics are insensitive to partition order



Latency Impact of Partitions



Sensor

Display Manager

Page Content Manager

Flight Manager

Flight Director

Request for new page

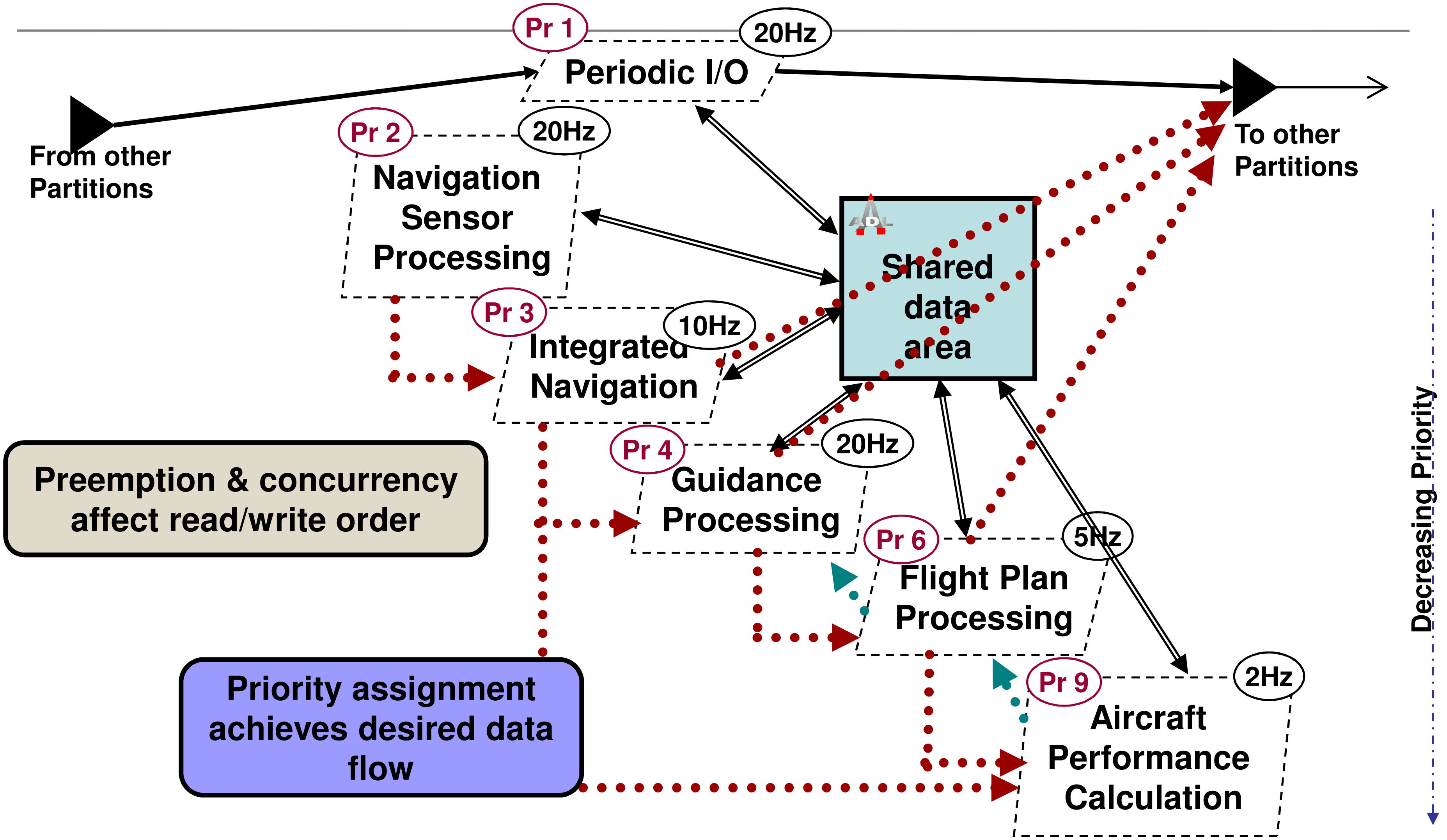
New page content

Lower bound on worst-case latency

Latency contribution:

Partition period per partition hop

Intended Data Flow in Task Architecture



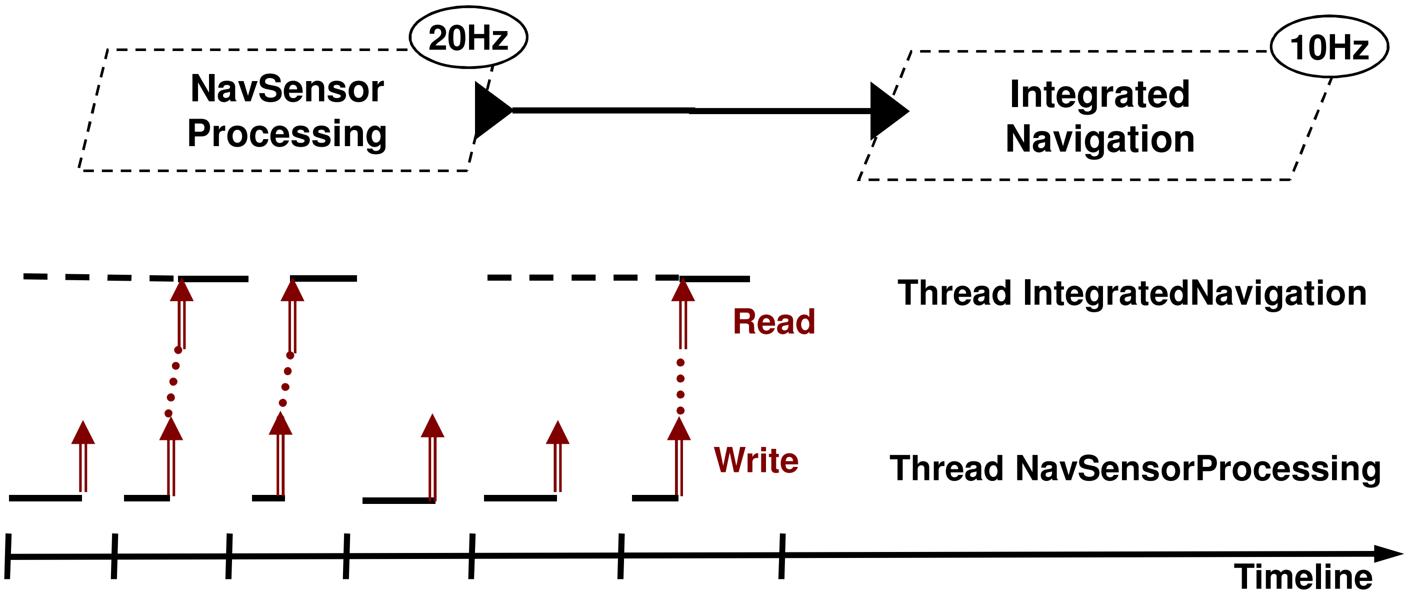
Preemption & concurrency affect read/write order

Priority assignment achieves desired data flow

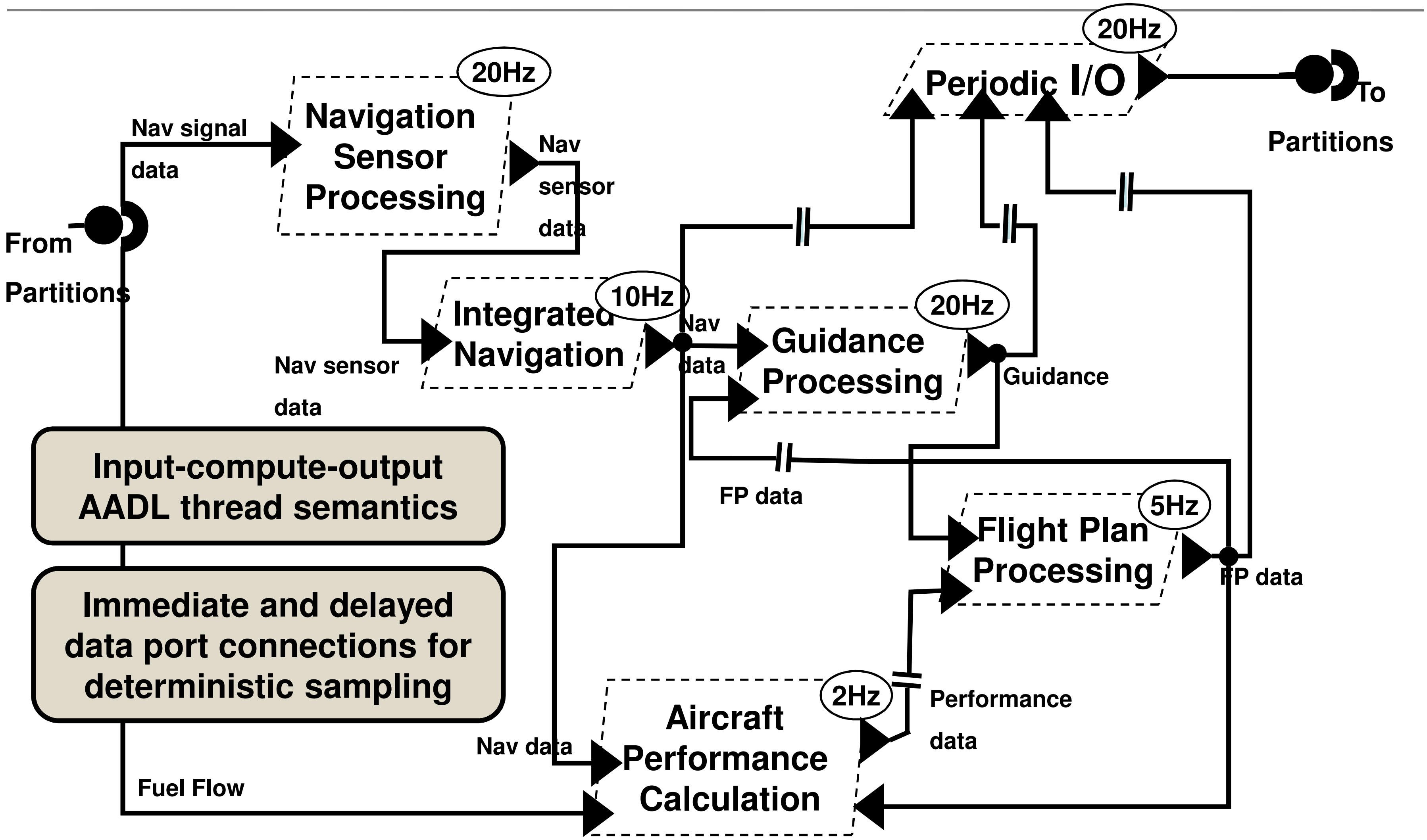
Frame-level Latency Jitter of Data Stream

Example: Non-deterministic downsampling

- Desired sampling pattern 2X: n, n+2, n+4 (2,2,2,…)
- Worst-case sampling pattern: n, n+1, n+4 (1,3,…)



Managed Latency Jitter through Deterministic Sampling



Input-compute-output
AADL thread semantics

Immediate and delayed
data port connections for
deterministic sampling

Rate Group Optimization

Logical threads to execute at a specific rate

Multiple logical threads to execute with the same rate

Placement of units with same rate in same operating system thread

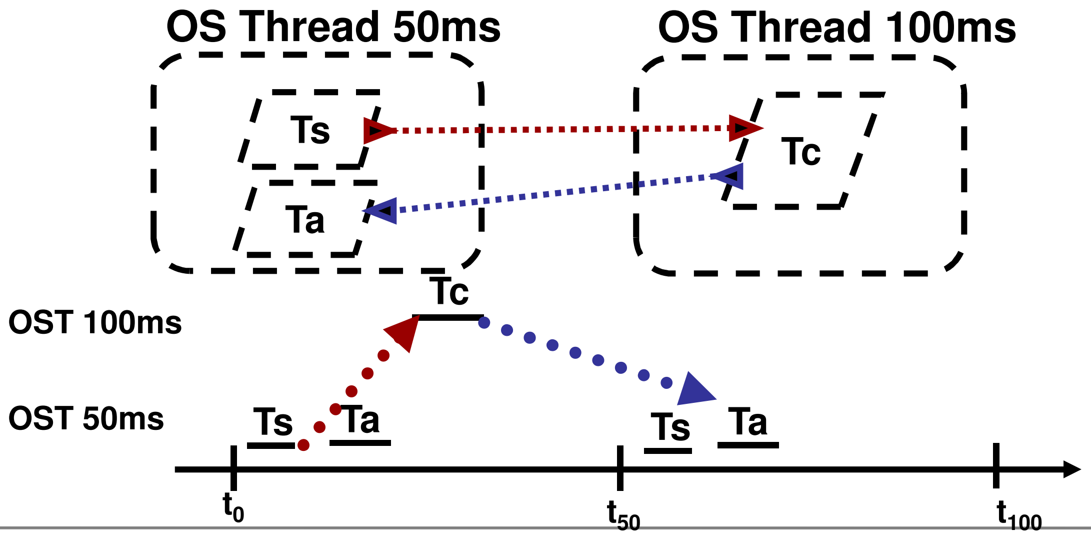
Reduced number of threads and context switches

Rate Group Order Can Affect Latency

Data flow from sensor T_s to control T_c to actuator T_a with mid-frame communication

Effect of rate groups: T_c to T_a becomes delayed

Occurs when pairwise immediate connections in opposite direction



Software-Based Latency Contributors

Execution time variation: algorithm, use of cache

Processor speed

Resource contention

Preemption

Legacy & shared variable communication

Rate group optimization

Protocol specific communication delay

Partitioned architecture

Migration of functionality

Fault tolerance strategy

Latency and Age of Data

Latency: the amount of time between a sensor reading and an output to an actuator based on the sensor reading

Age: amount of time that has passed since the sensor reading

Age Contributors

- Oversampling
- Missing sensor readings
- Failed processing
- Missed deadlines

Outline

Multiple aspects of system validation

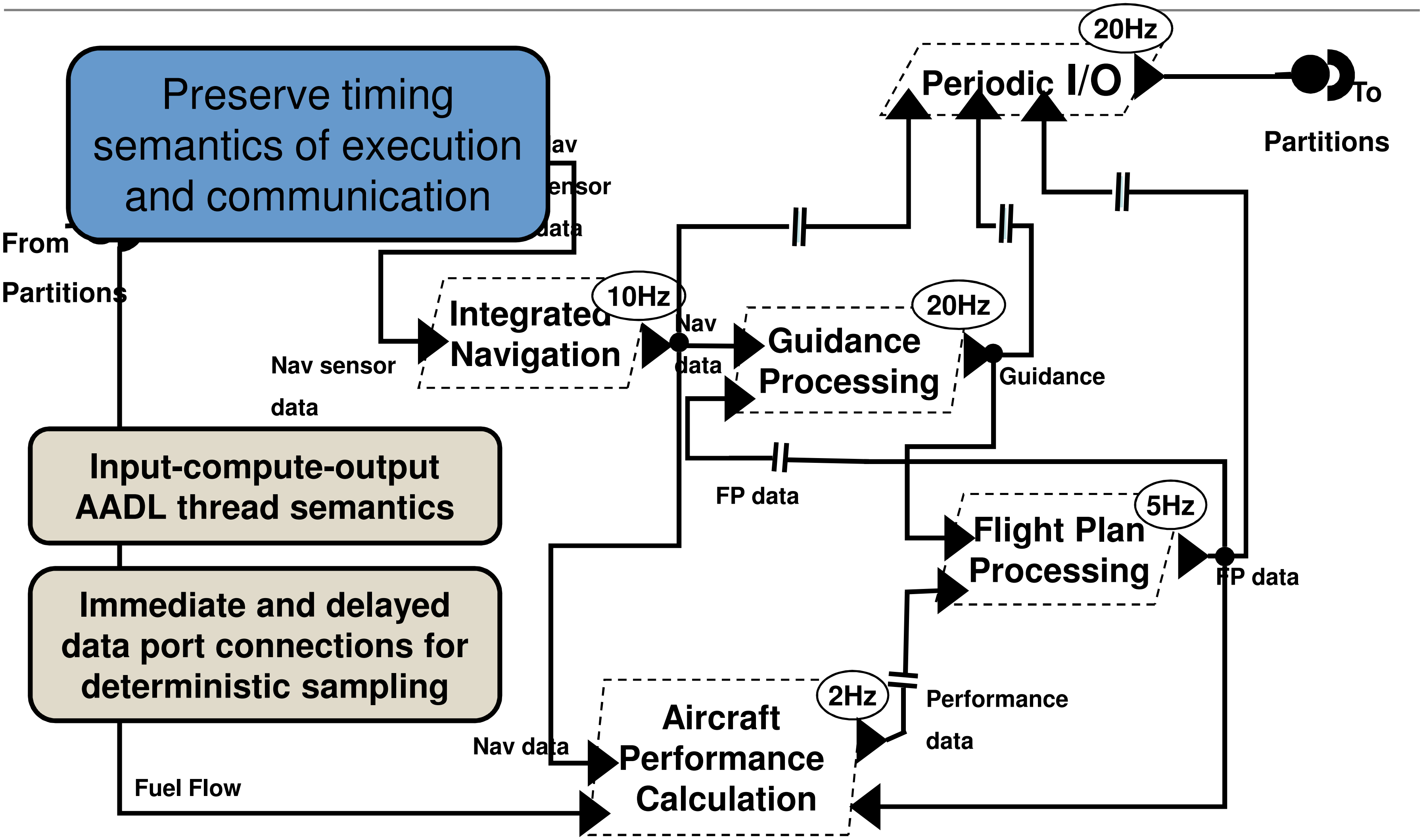
System & software engineers working together

Multi-fidelity model-based analysis

Property preserving transformations

Conclusions

Efficient Runtime System Generation

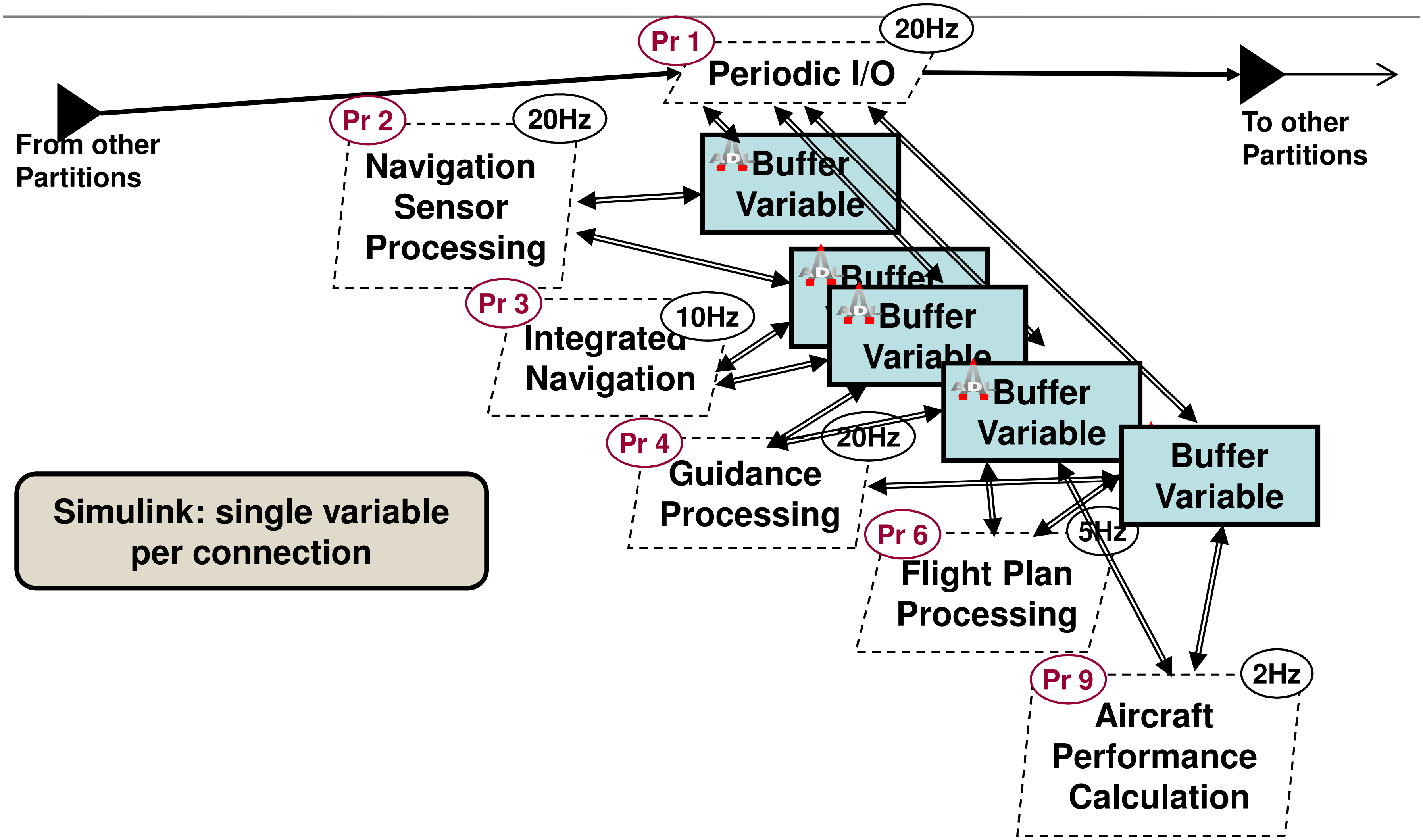


Preserve timing semantics of execution and communication

Input-compute-output AADL thread semantics

Immediate and delayed data port connections for deterministic sampling

Will This Implementation Work?



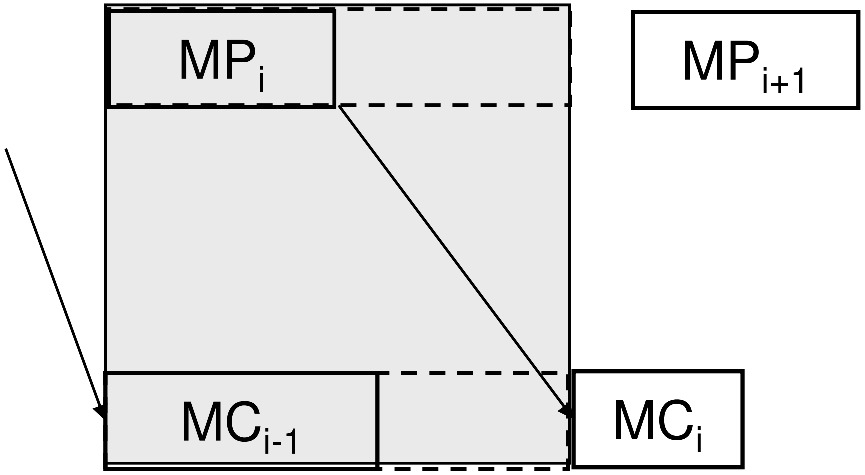
Simulink: single variable per connection

Overlapping Message Lifespan

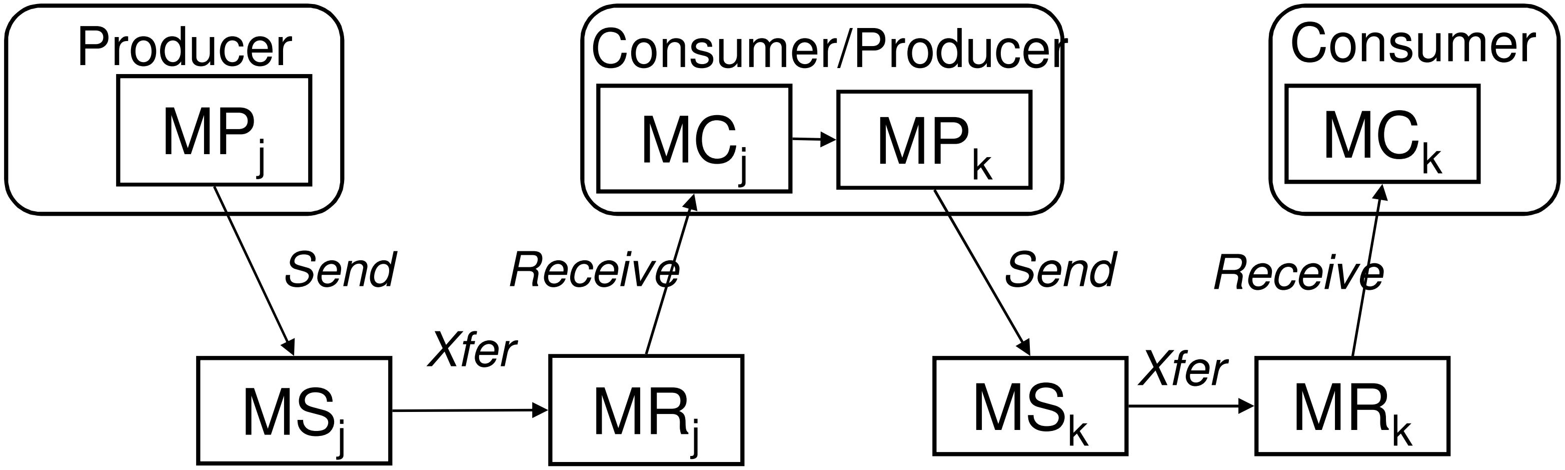
Periodic thread MP and MC

MP ->> MC

Need for double buffering



Optimization of General Port Buffer Model



- Send/receive with or without copy
- Transfer with or without copy
- Processing with or without copy

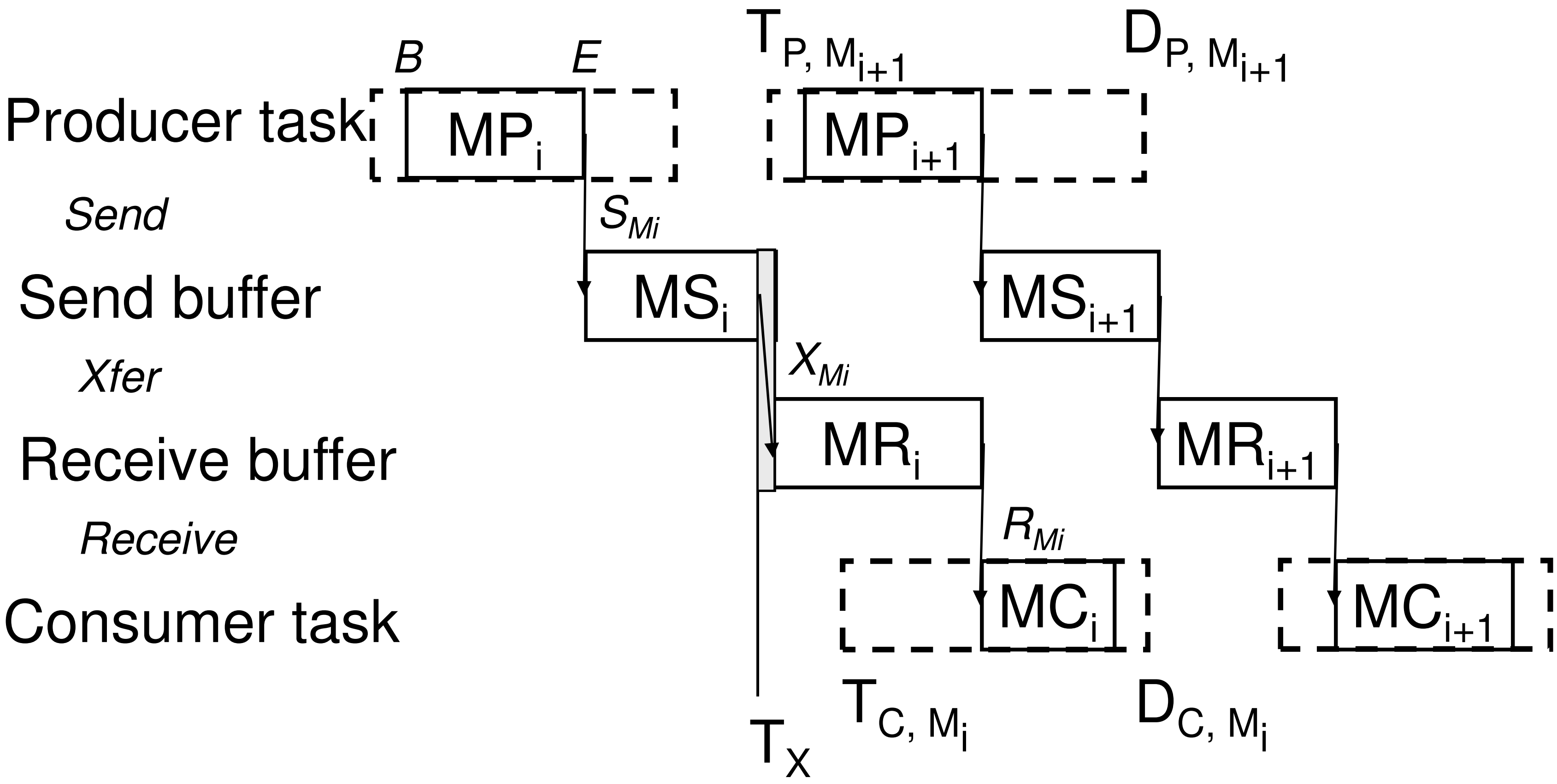
MP: producer copy

MS: send copy

MR: receive copy

MC: consumer copy

Message Streaming Lifespan Framework



Message Lifespan Properties

MC input-compute-output guarantee

$T_{C,\ M_i} \le R_{M_i} = B_{MC_i} \le \mathbf{E_{MC_i}} \le T_{C,\ M_{i+1}} \le \mathbf{R_{m_{i+1}}}$

Message operation ordering condition

$S_{M_i} < X_{M_i} < R_{M_i}$

MP bounded by producer dispatches

$T_{P,\ M_i} \le B_{MP_i} \le E_{MP_i} = S_{M_i} \le T_{P,\ M_{i+1}}$

MS bounded by sends and transfer

$S_{M_i} = B_{MS_i} \le X^{*}_{M_i} \le E_{MS_i} < S_{M_{i+1}}$

MR bounded by transfers and receive

$X^{**}_{M_i} \le B_{MR_i} \le E_{MR_i} = R^{***}_{M_i} < X_{M_{i+1}}$

* Completion of transfer

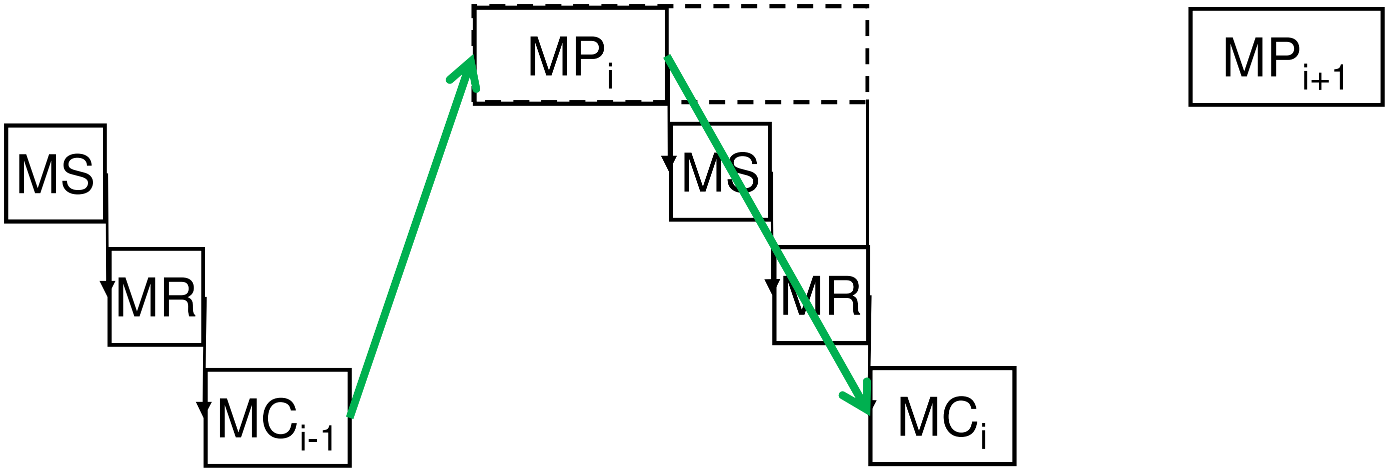
** Start of transfer

*** Latest of multiple receivers

Sequential Execution of Periodic Tasks

$(\tau_P ; \tau_C)^*$

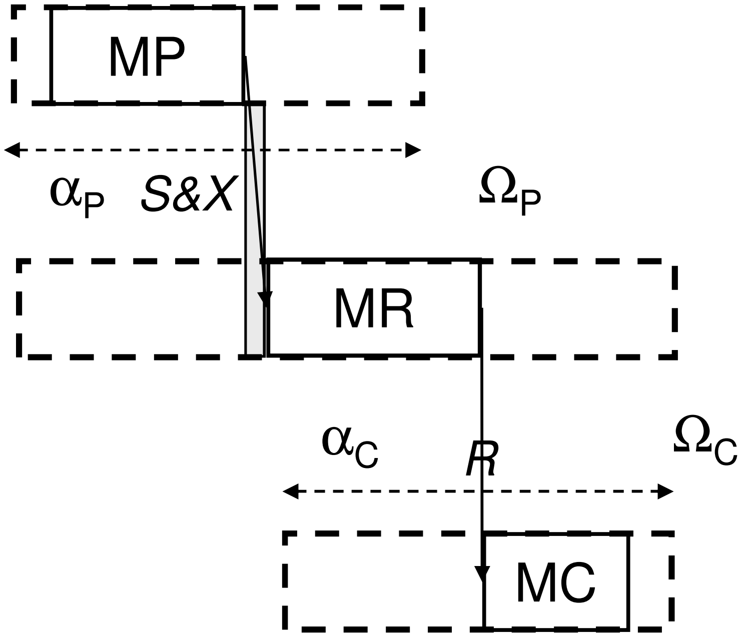
Collapse to single buffer



Application-based Send and Receive (ASR)

$(\tau_P \mid \tau_C)^*$

3 buffers



$T_P \leq \alpha_P \leq S \leq \Omega_P \leq D_P$

$T_C \leq \alpha_C \leq R \leq \Omega_C \leq D_C$

α : actual execution start time

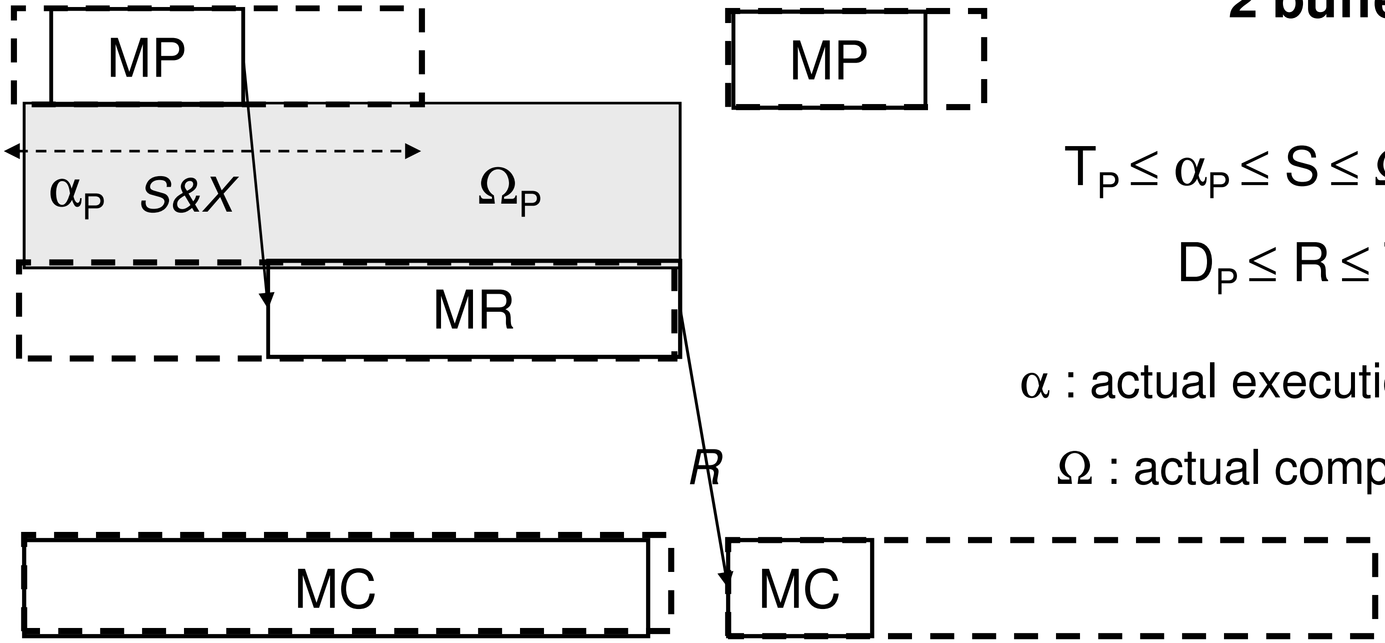
Ω : actual completion time

$\alpha_P - \Omega_P \cap \alpha_C - \Omega_C \neq \varnothing \Rightarrow$ non-deterministic S/R order

Dispatch-based Send and Receive (DSR)

$(\tau_P \mid \tau_C)^*$

2 buffers



$T_P \le \alpha_P \le S \le \Omega_P \le D_P$

$D_P \le R \le T_C$

α : actual execution start time

Ω : actual completion time

$\alpha_P - \Omega_P \cap D_P - T_C = \emptyset \Rightarrow$ deterministic S/R

Buffer Optimization Considerations

Periodic & aperiodic task dispatch

Send and receive execution

- As part of application (ASR)
- As part of task dispatch/completion (DSR)

Task execution order

- Concurrent: $\tau_C \mid \tau_P$
- Atomic non-deterministic: $\tau_C \neq \tau_P$
- Ordered: $\tau_C ; \tau_P$ or $\tau_P ; \tau_C$

Message transfer

- Immediate to consumer (IMT)
- Direct to delayed consumer (DMT)
- Period-delayed to consumer (PMT)

Periodic Task Communication Summary

Periodic	ASR		DSR		DMT
Same period	IMT	PMT	IMT	PMT	
$\tau_P ; \tau_C$	MF:1B	PD:2B $S \vee X \vee R$	PD:2B R	PD:2B $S \vee X/R$	MF:1B
$\tau_C ; \tau_P$	PD:1B	PD:1B	PD:1B	PD:1B	PD:1B
$\tau_P \neq \tau_C$	ND:1B	PD:2B X	PD:2B R	PD:2B X/R	ND:1B
$\tau_P \mid \tau_C$	ND:3B S/X_C R_C	PD:2B X	PD:2B R	PD:2B X/R	NDI:2B $S/X/R_C$

MF: Mid-Frame
PD: Period Delay
ND: Non-Deterministic
NDI: No Data Integrity

1B: Single buffer
2B: Two buffers
3B: Three buffers
4B: Four buffers

S, X, R : data copy
S/X : IMT combined send/xfer
S/X/R : DMT combined S, X, R
X/R: DSR/PMT combined X, R
o1$\vee$o2 : One operation copy

Outline

Multiple aspects of system validation

System & software engineers working together

Multi-fidelity model-based analysis

Property preserving transformations

Conclusions

Predictable Model-based Engineering

Reduce the risks

- Analyze system early and throughout life cycle
- Understand system wide impact
- Validate assumptions across system

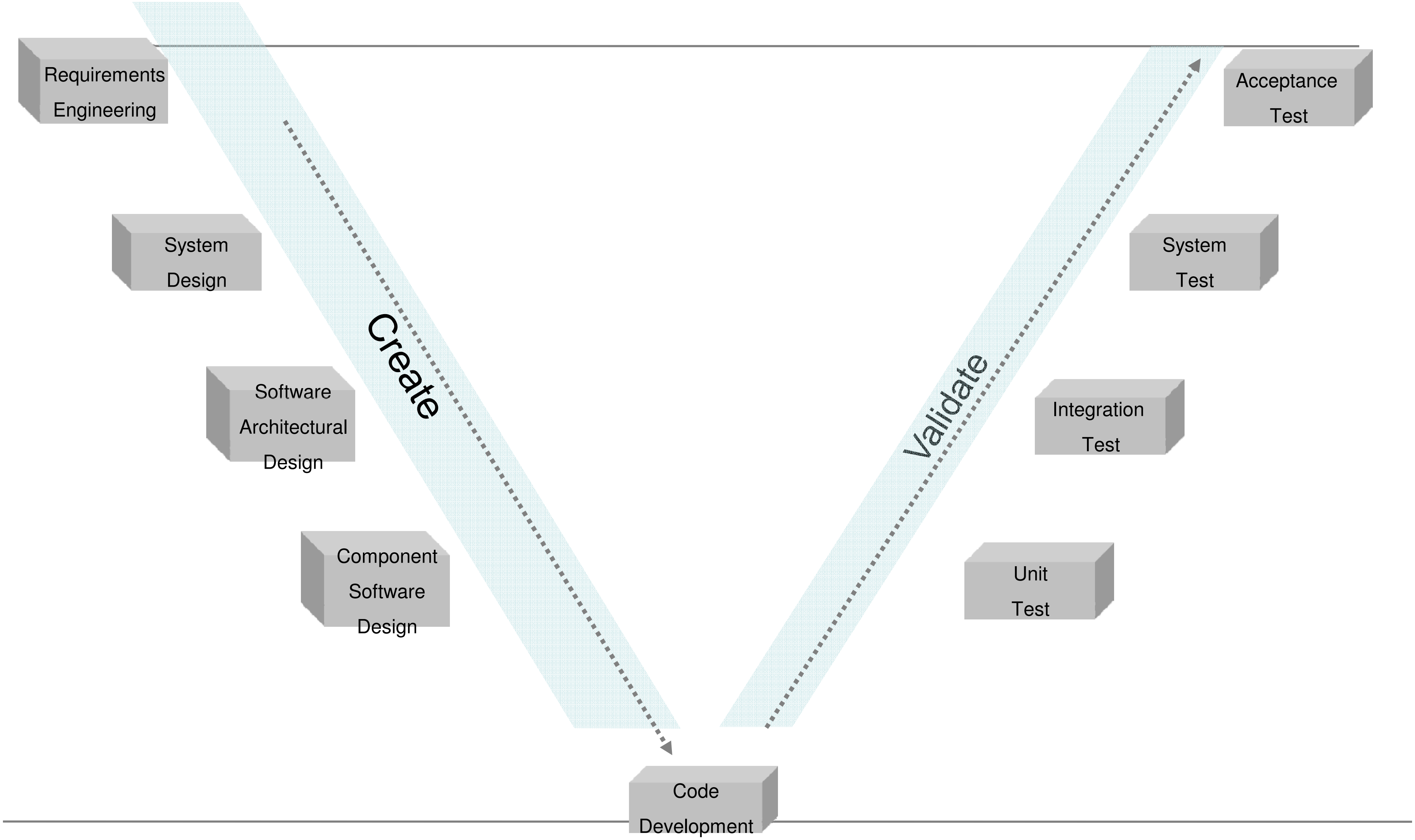
Increase the confidence

- Validate models to complement integration testing
- Validate model assumptions in operational system
- Evolve system models in increasing fidelity

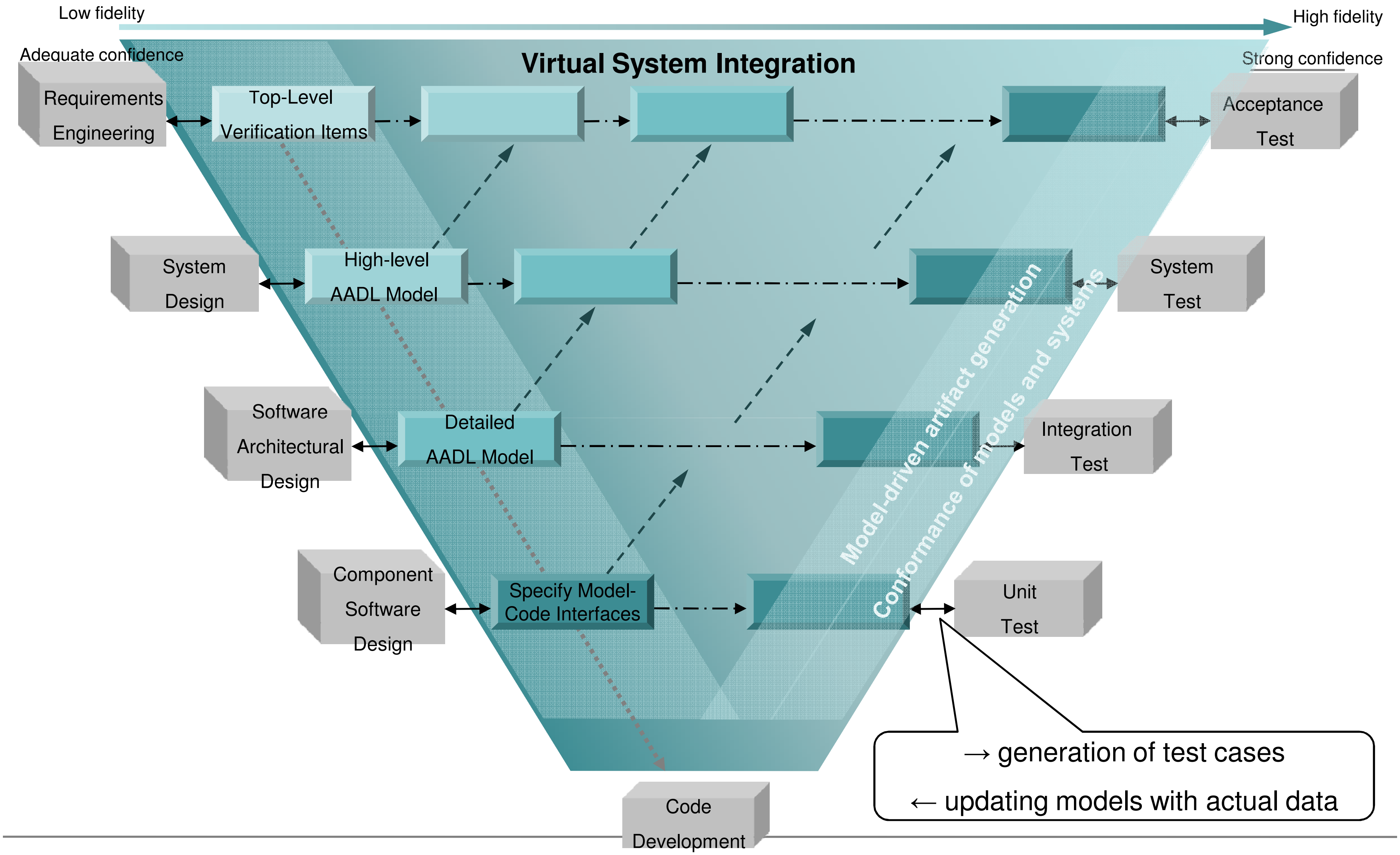
Reduce the cost

- Fewer system integration problems
- Fewer validation steps through use of validated generators

Traditional Development Model



Benefits of Predictive Architecting



Low fidelity

High fidelity

Adequate confidence

Strong confidence

Virtual System Integration

Requirements Engineering

Top-Level Verification Items

Acceptance Test

System Design

High-level AADL Model

System Test

Software Architectural Design

Detailed AADL Model

Integration Test

Component Software Design

Specify Model-Code Interfaces

Unit Test

Code Development

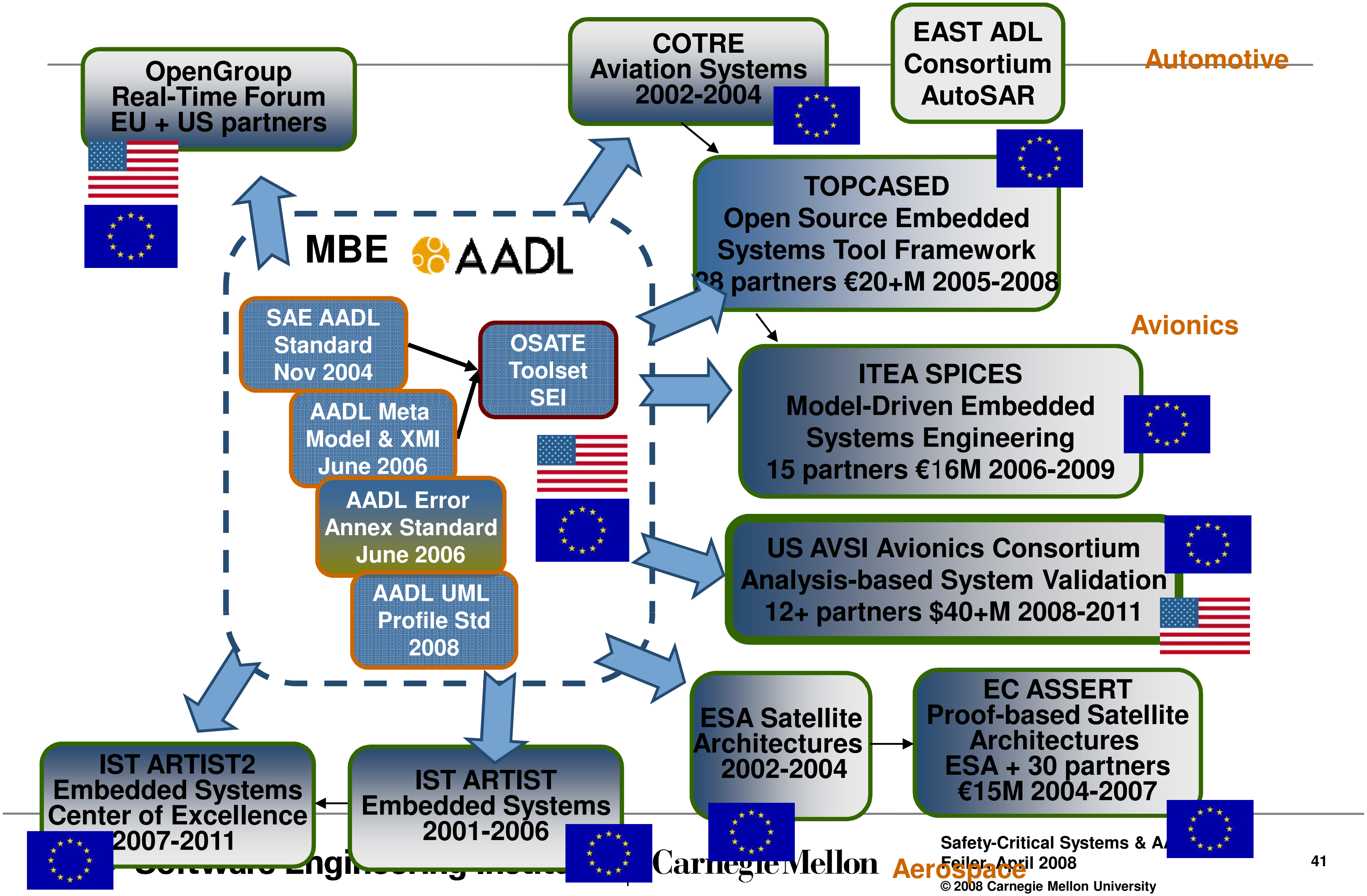
Model-driven artifact generation

Conformance of models and systems

→ generation of test cases

← updating models with actual data

Industrial Embedded Systems Initiatives



Automotive

Avionics

Aerospace

MBE AADL

SAE AADL Standard Nov 2004

AADL Meta Model & XMI June 2006

AADL Error Annex Standard June 2006

AADL UML Profile Std 2008

OSATE Toolset SEI

OpenGroup Real-Time Forum EU + US partners

COTRE Aviation Systems 2002-2004

EAST ADL Consortium AutoSAR

TOPCASED Open Source Embedded Systems Tool Framework 28 partners €20+M 2005-2008

ITEA SPICES Model-Driven Embedded Systems Engineering 15 partners €16M 2006-2009

US AVSI Avionics Consortium Analysis-based System Validation 12+ partners $40+M 2008-2011

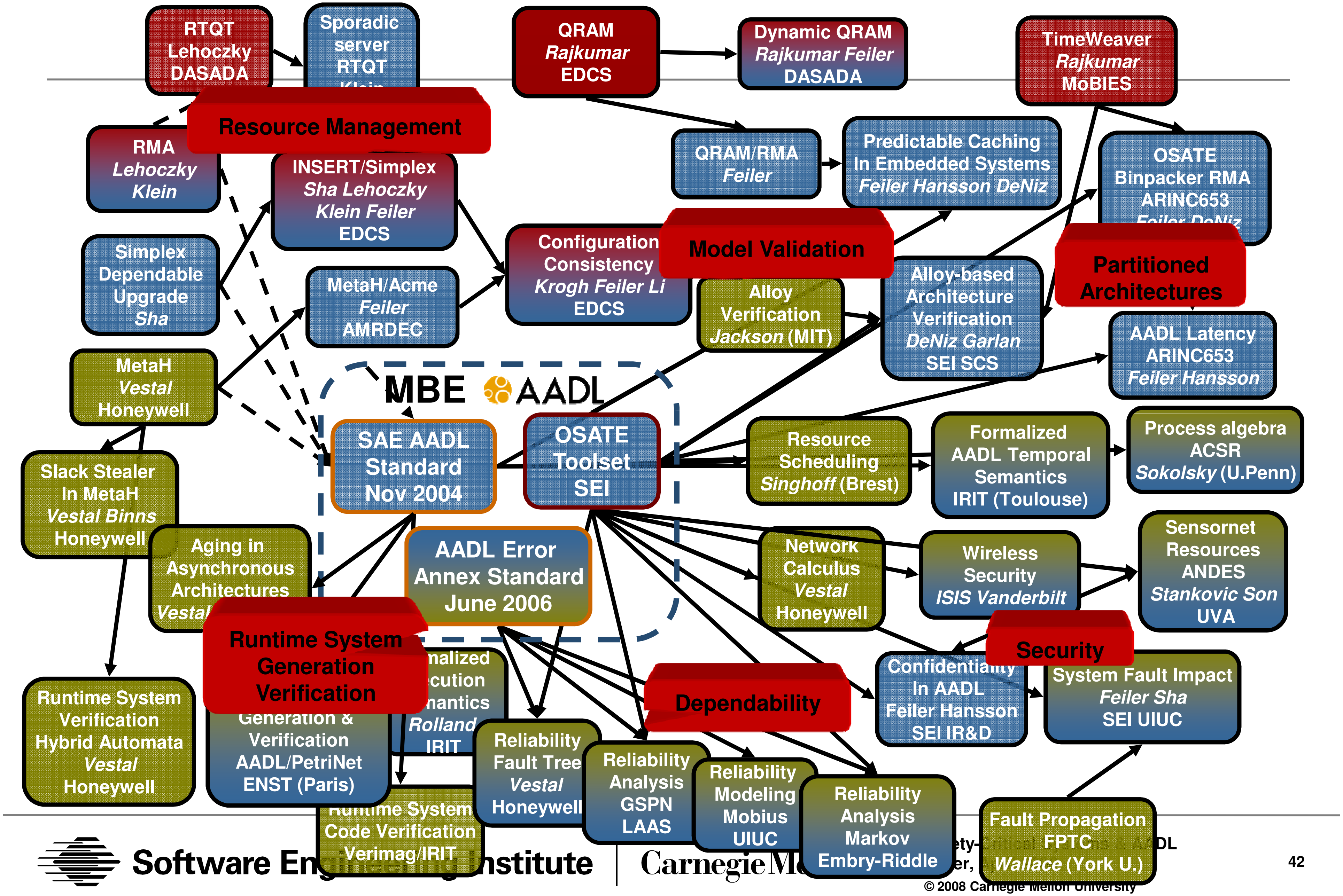
ESA Satellite Architectures 2002-2004

EC ASSERT Proof-based Satellite Architectures ESA + 30 partners €15M 2004-2007

IST ARTIST Embedded Systems 2001-2006

IST ARTIST2 Embedded Systems Center of Excellence 2007-2011

A Research Transition Platform



RTQT
Lehoczky
DASADA

Sporadic server
RTQT
Klein

QRAM
Rajkumar
EDCS

Dynamic QRAM
Rajkumar Feiler
DASADA

TimeWeaver
Rajkumar
MoBIES

Resource Management

RMA
Lehoczky Klein

INSERT/Simplex
Sha Lehoczky Klein Feiler
EDCS

QRAM/RMA
Feiler

Predictable Caching In Embedded Systems
Feiler Hansson DeNiz

OSATE
Binpacker RMA
ARINC653
Feiler DeNiz

Simplex Dependable Upgrade
Sha

MetaH/Acme
Feiler
AMRDEC

Configuration Consistency
Krogh Feiler Li
EDCS

Model Validation

Alloy Verification
Jackson (MIT)

Alloy-based Architecture Verification
DeNiz Garlan
SEI SCS

Partitioned Architectures

AADL Latency
ARINC653
Feiler Hansson

MetaH
Vestal
Honeywell

MBE AADL

SAE AADL Standard
Nov 2004

OSATE Toolset
SEI

Slack Stealer In MetaH
Vestal Binns
Honeywell

Resource Scheduling
Singhoff (Brest)

Formalized AADL Temporal Semantics
IRIT (Toulouse)

Process algebra
ACSR
Sokolsky (U.Penn)

Aging in Asynchronous Architectures
Vestal

AADL Error Annex Standard
June 2006

Network Calculus
Vestal
Honeywell

Wireless Security
ISIS Vanderbilt

Sensornet Resources
ANDES
Stankovic Son
UVA

Runtime System Generation Verification

Security

Runtime System Verification
Hybrid Automata
Vestal
Honeywell

Generation & Verification
AADL/PetriNet
ENST (Paris)

malized
cution
mantics
Rolland
IRIT

Dependability

Confidentiality In AADL
Feiler Hansson
SEI IR&D

System Fault Impact
Feiler Sha
SEI UIUC

Reliability Fault Tree
Vestal
Honeywell

Reliability Analysis
GSPN
LAAS

Reliability Modeling
Mobius
UIUC

Reliability Analysis
Markov
Embry-Riddle

Runtime System Code Verification
Verimag/IRIT

Fault Propagation
FPTC
Wallace (York U.)

Software Engineering Institute | Carnegie Mellon

Peter H Feiler

phf@sei.cmu.edu

at ENST until June 9, 2008