USC Viterbi
School of Engineering
Center for Cyber-Physical Systems and the Internet of Things

From Electronic Design Automation to Cyber-Physical System Design Automation: A Tale of Platforms and Contracts

Pierluigi Nuzzo
Ming Hsieh Department of Electrical and Computer Engineering
University of Southern California, Los Angeles
nuzzo@usc.edu

In Honor of Alberto Sangiovanni-Vincentelli

International Symposium on Physical Design, San Francisco, April 16, 2019

Cyber-Physical System Design: What Can Go Wrong?

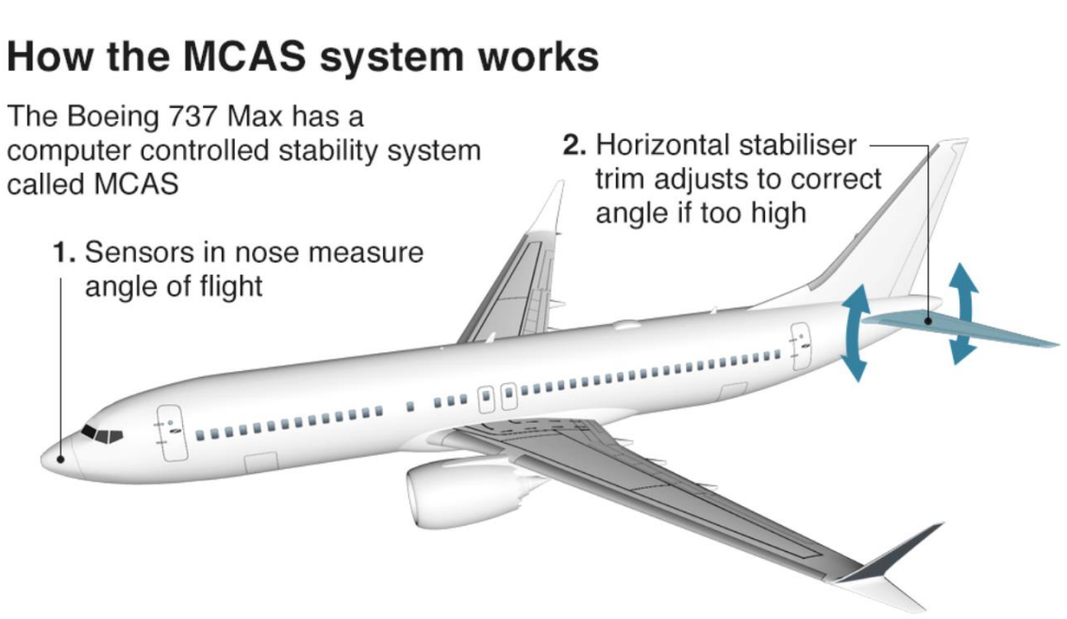
BBC NEWS

Pilots of the crashed Ethiopian Airlines Boeing 737 Max were unable to prevent the plane repeatedly nosediving despite following procedures, an initial report has found.

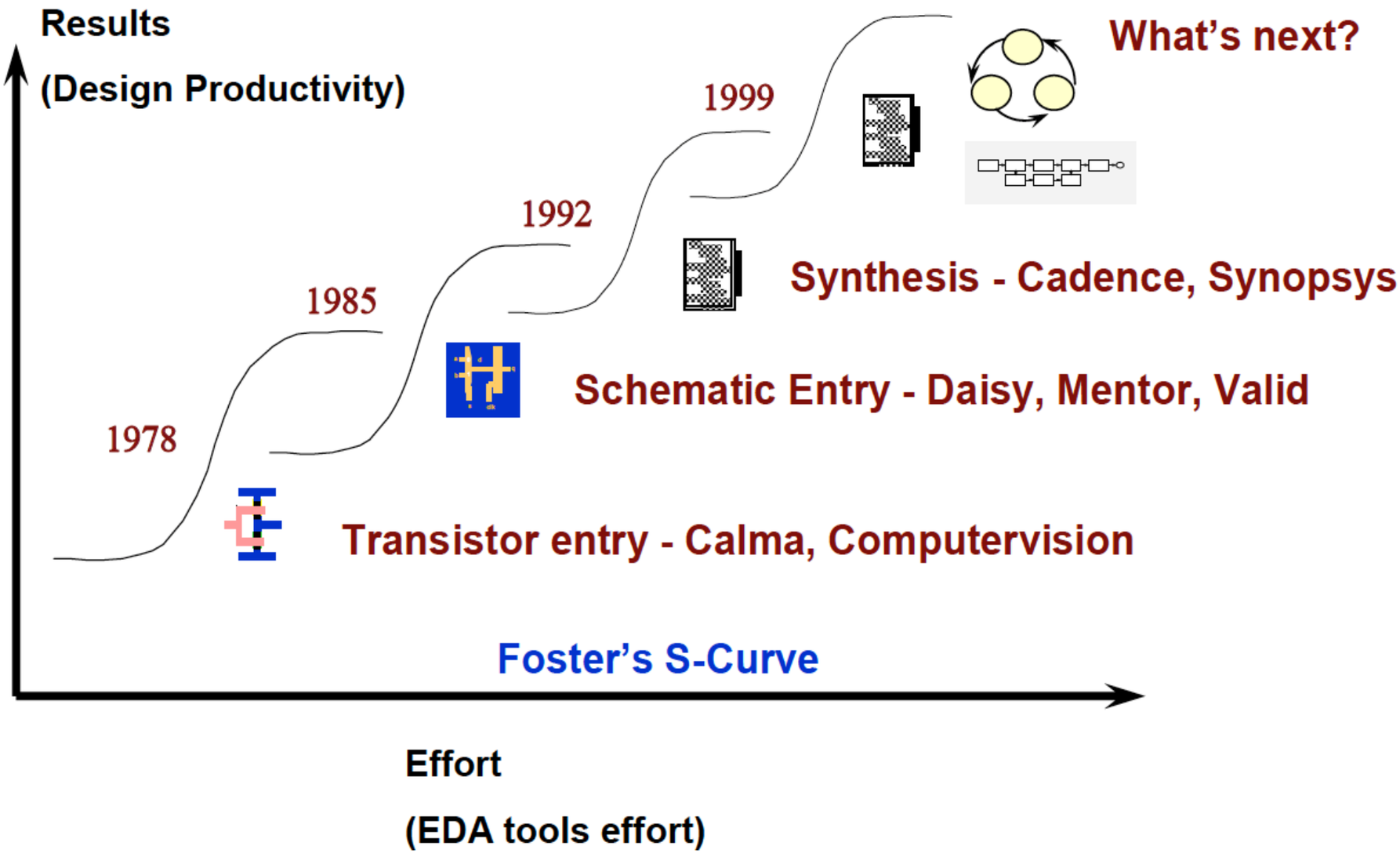
The captain and first officer followed safety procedures recommended by Boeing. But **they couldn't stop the aircraft going into a fatal dive** shortly after take off from Addis Ababa on 10 March, the report by Ethiopian investigators said. All 157 people on board were killed.

Aviation authorities grounded the entire global fleet of 737 Max aircraft in March after two fatal crashes in five months.

The Ethiopian Airlines crash followed a Lion Air crash in Indonesia in October, which left 189 dead.



The Quest for the Next Level of Abstraction: System Level Design



Courtesy: A. Sangiovanni-Vincentelli

Taming Dr. Frankenstein: Contract-Based Design for Cyber-physical Systems

Alberto Sangiovanni-Vincentelli[‖], Werner Damm[**],
and Roberto Passerone[‡],

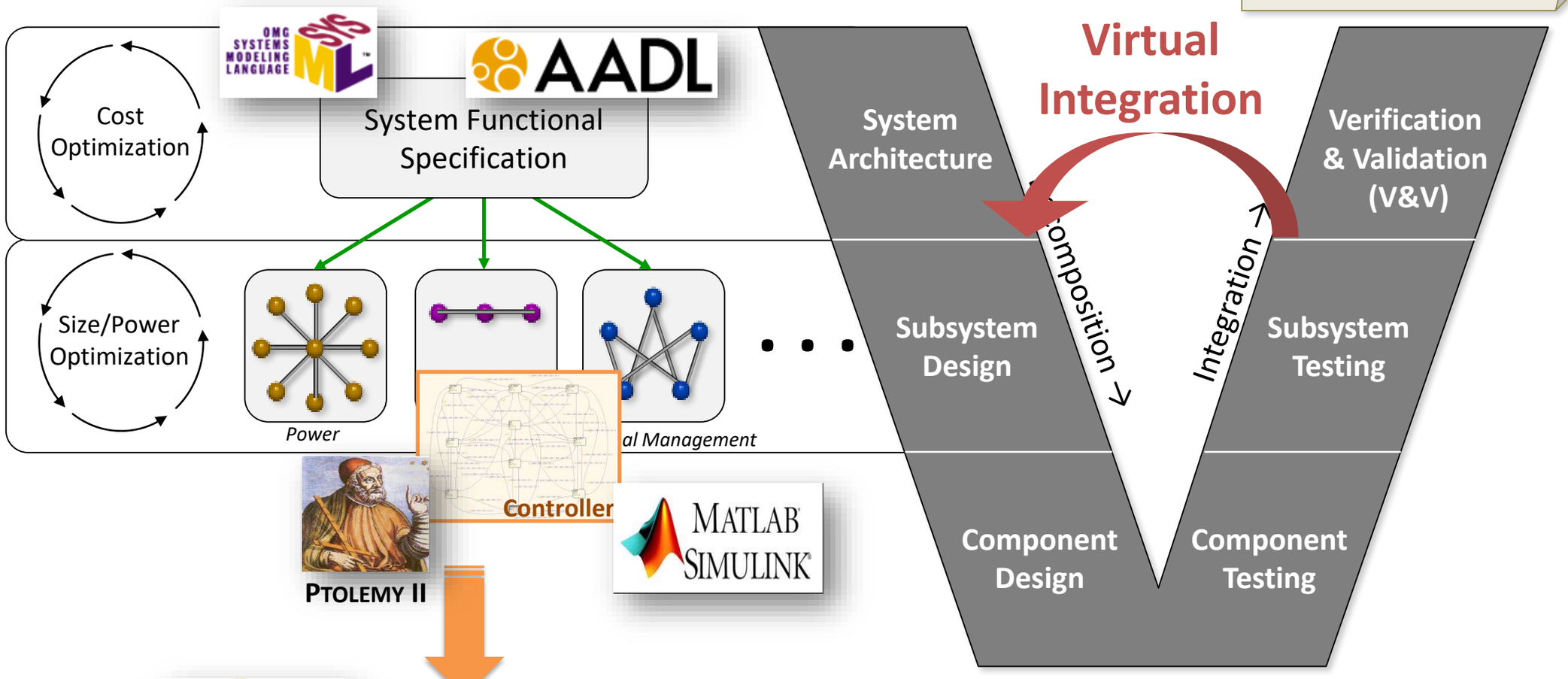
- Platform-Based Design
- Contracts
- Applications
- What's next?

Cyber-Physical System Design: State of the Art

*Conventional V&V techniques do not **scale** to highly complex or adaptable systems*

"Let's Get Physical: Computer Science Meets Systems," ETAPS Workshop, 2014

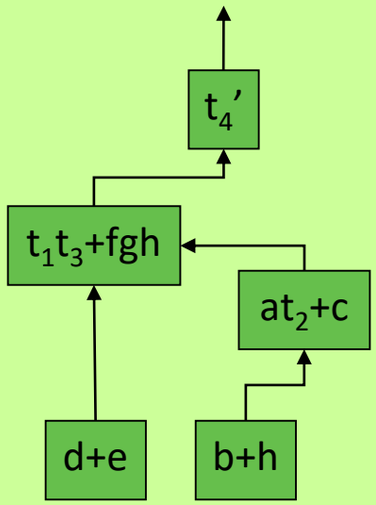
***Experienced architects** must rely on accrued **knowledge** and **heuristics** to take risky decisions*



*A large number of **poorly integrated** languages and tools*

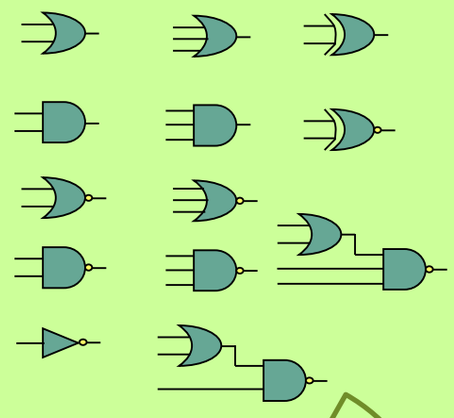
Learning from Logic Synthesis

High level function model



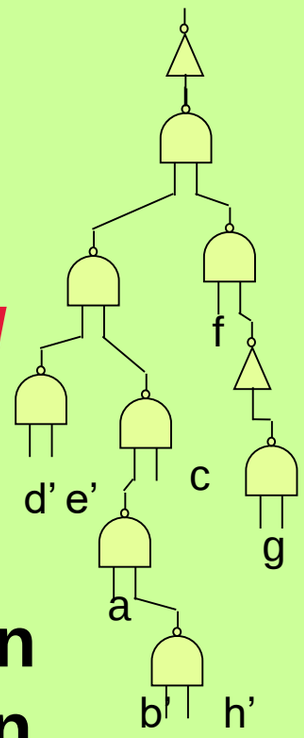
- Separation of function and architecture
- Common language for functional and architectural level netlists (Boolean logic, NAND2 gate)
- Automatic mapping

Gate library (platform)

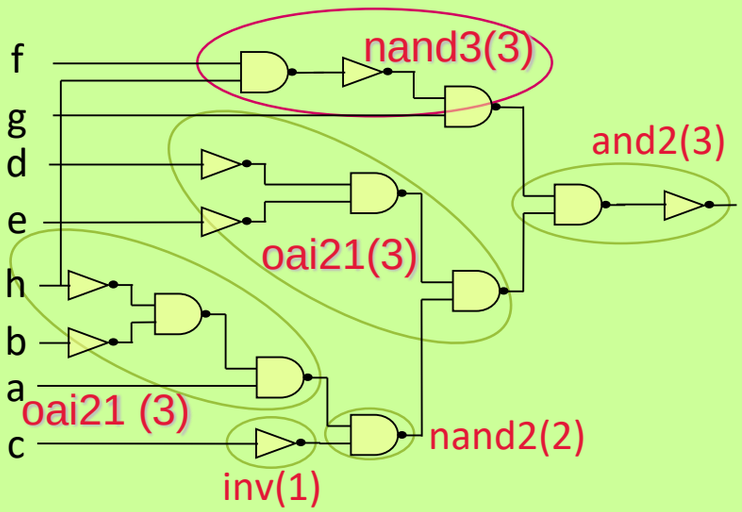


Technology Mapping (covering)

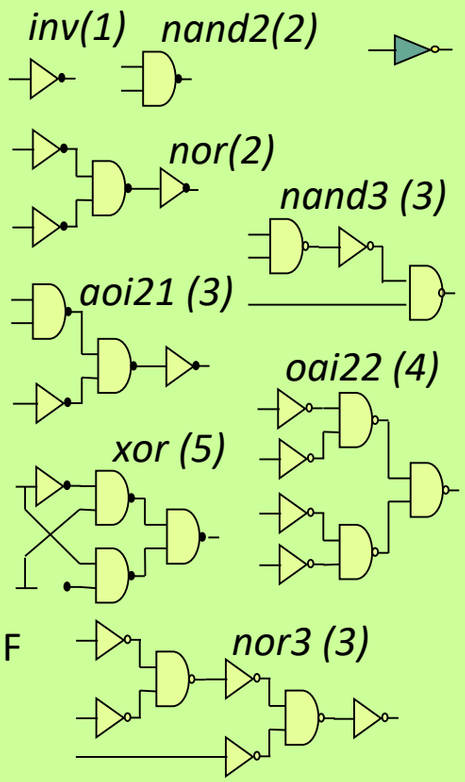
restructuring



Function model in netlist



Mapped design



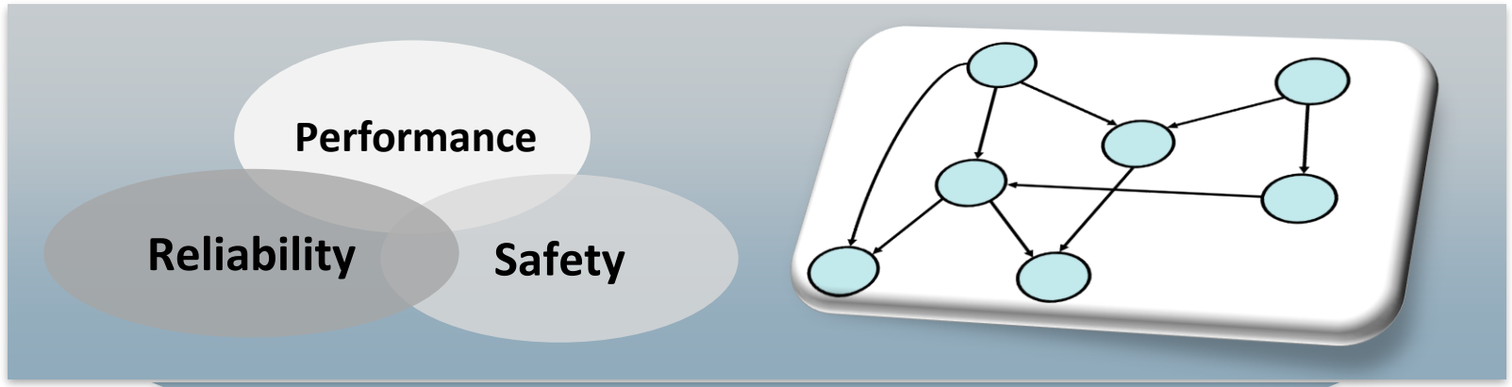
restructuring

Gate library in netlist

Courtesy: A. Sangiovanni-Vincentelli

Platform-Based Design

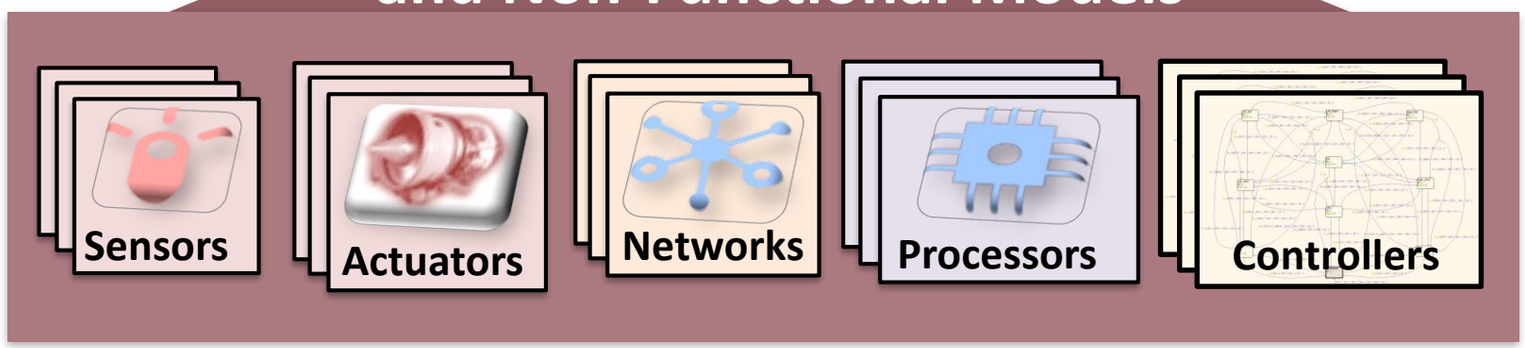
Application Space: System Specification



System Requirements

Synthesis (Optimization)

Behavioral and Non-Functional Models



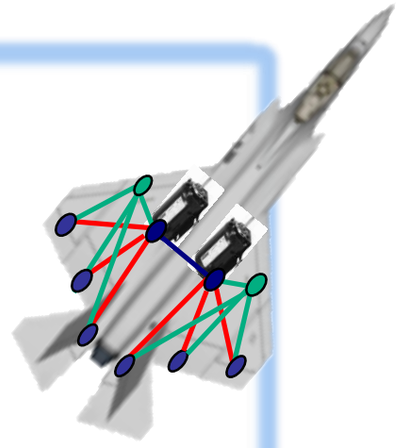
Implementation Space: Platform Library

[A. Sangiovanni-Vincentelli and A. Ferrari, '90]

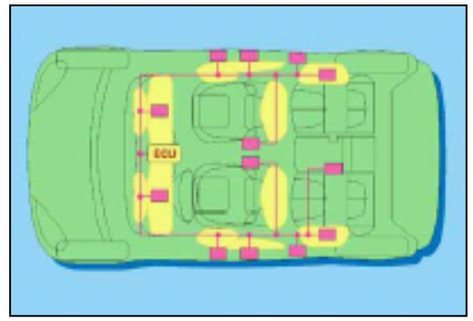
Quo Vadis, SLD? Reasoning About the Trends and Challenges of System Level Design

Recognizing common requirements for co-design of hardware and software in diverse systems may lead to productivity gains, lower costs and first-pass design success.

By Alberto Sangiovanni-Vincentelli, *Fellow IEEE*

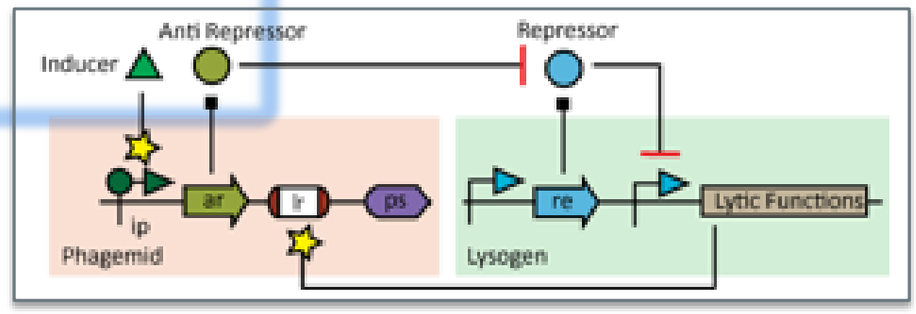


Avionics

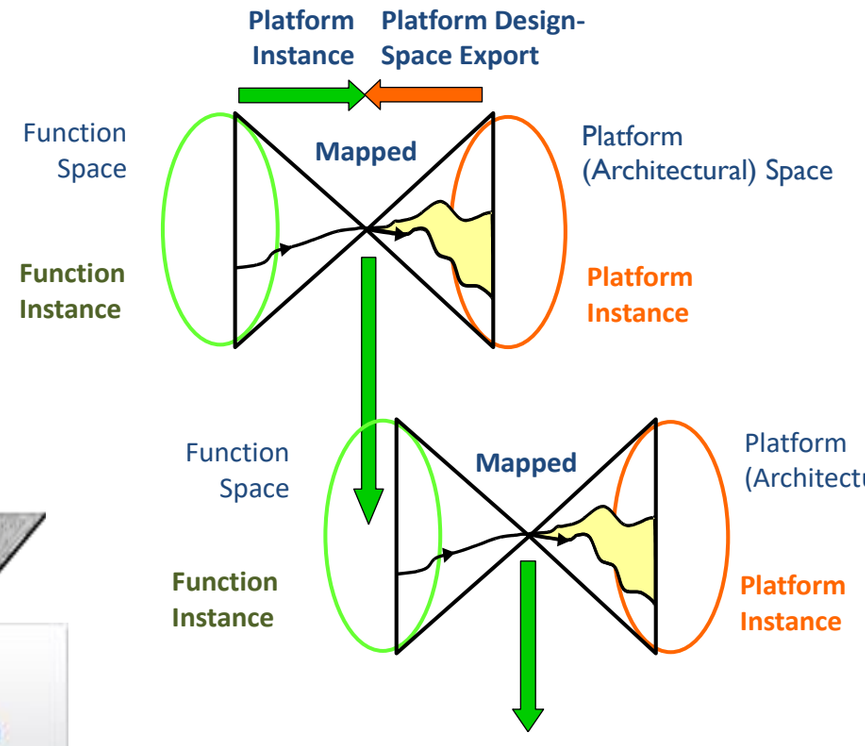


Automotive

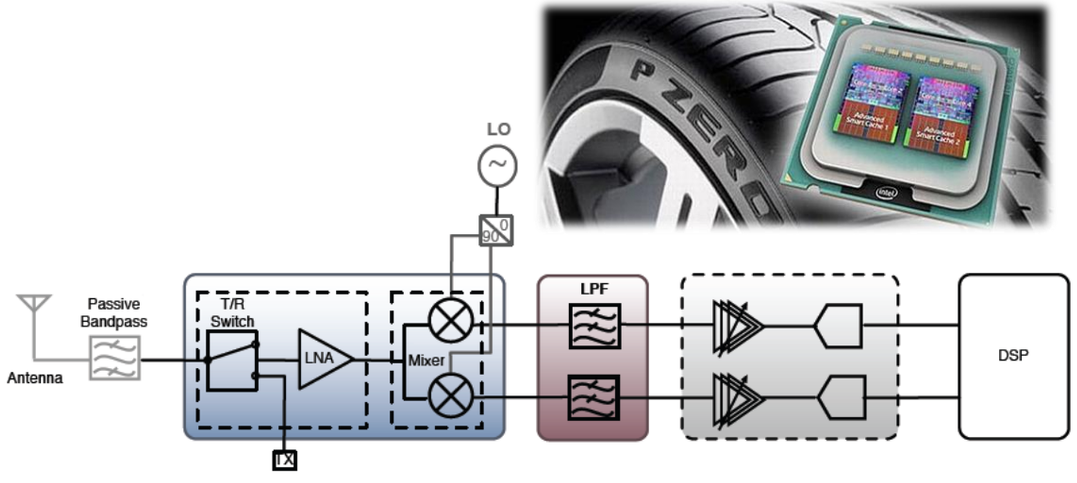
Smart Buildings



Synthetic Biology



ASV Triangles



Mixed-Signal Systems on Chip

Platform-Based Design With Contracts

Requirement Formalization

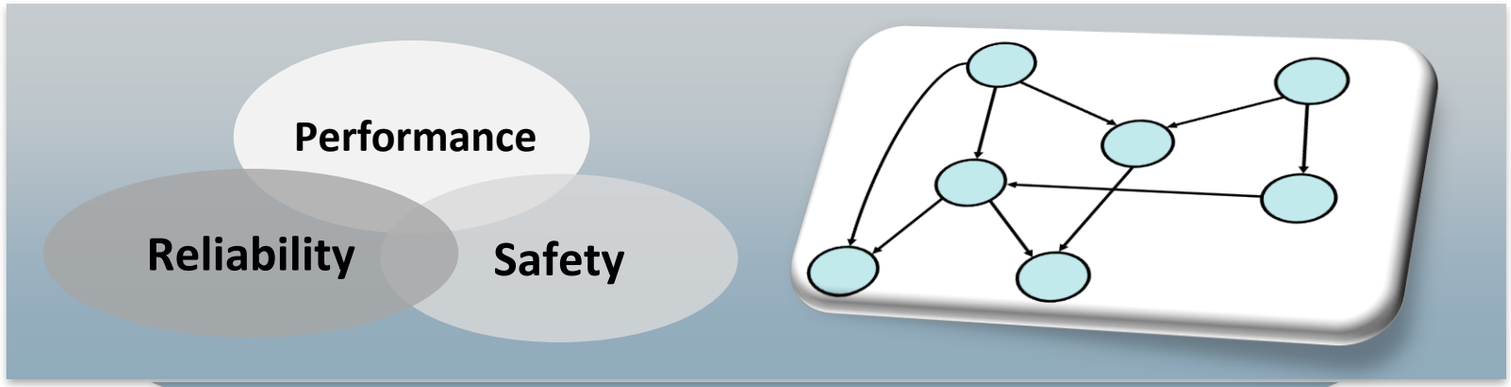
Refinement Rules

Contracts

Abstraction Rules

Composition Rules

Application Space: System Specification

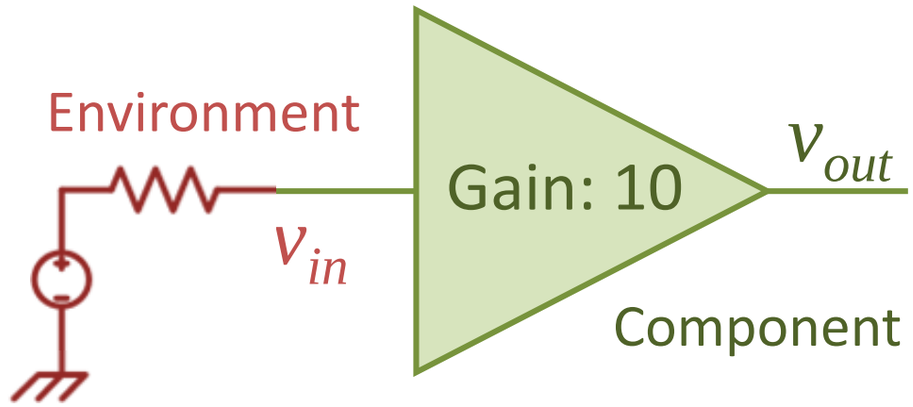


Implementation Space: Platform Library

Assume/Guarantee (A/G) Contracts

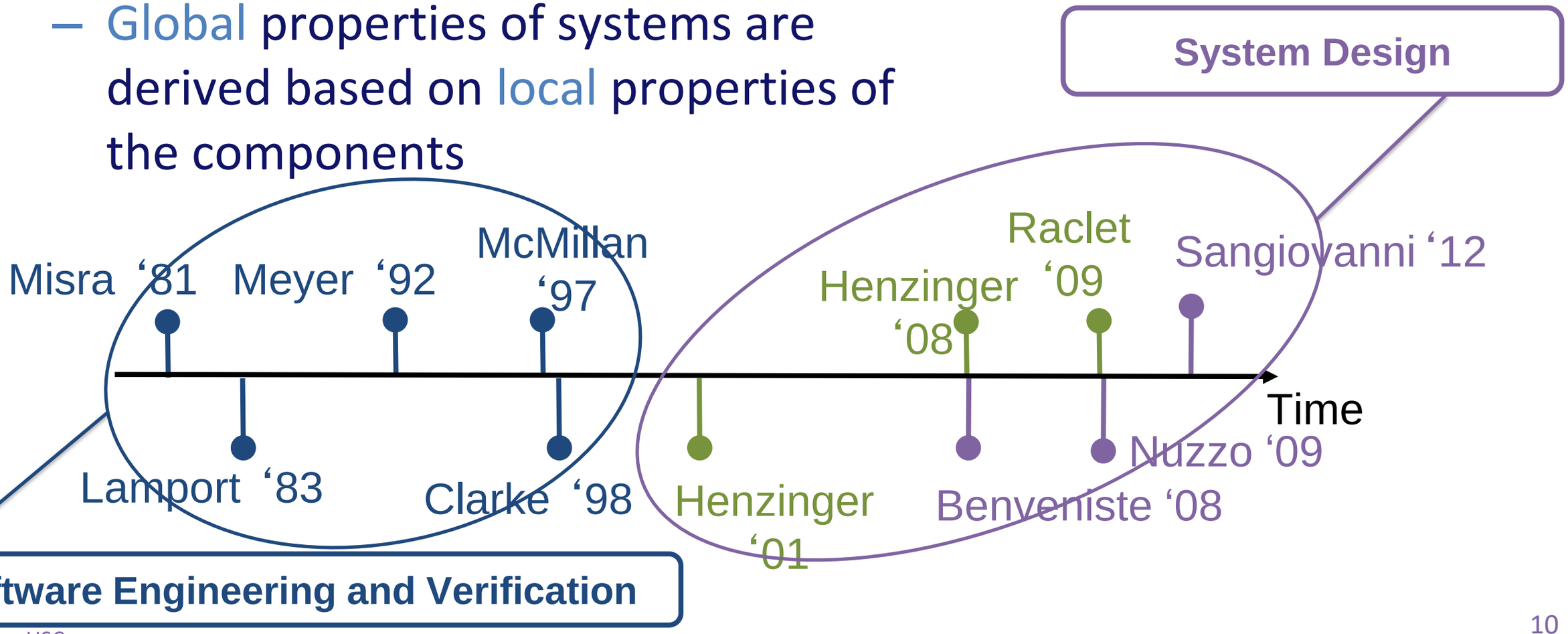
Contracts are Assume-Guarantee pairs

- Component properties are guaranteed under a set of assumptions on the environment
- Global properties of systems are derived based on local properties of the components

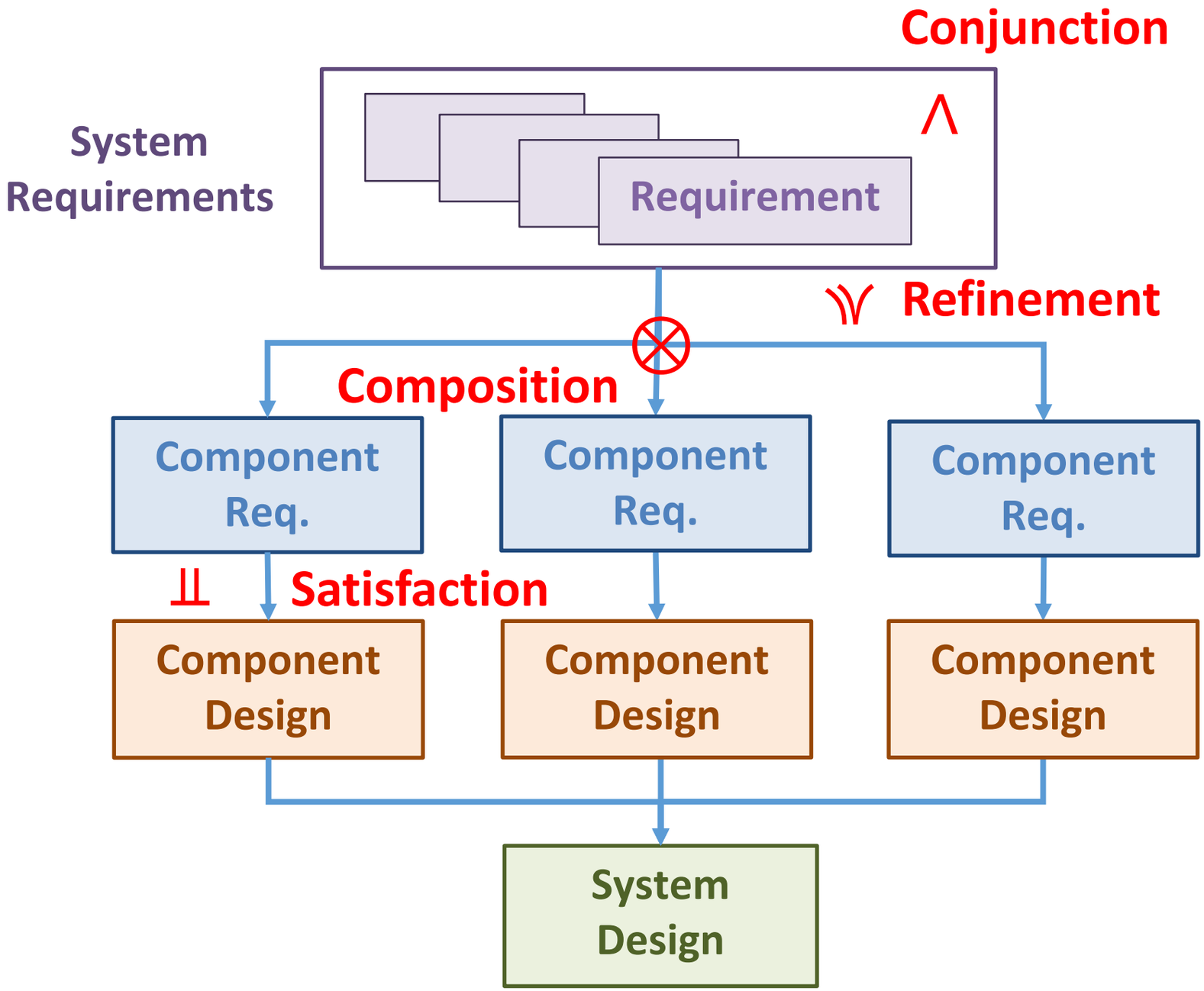


Assumptions: $|v_{in}| \leq 2$

Guarantees: $v_{out} = 10v_{in}$

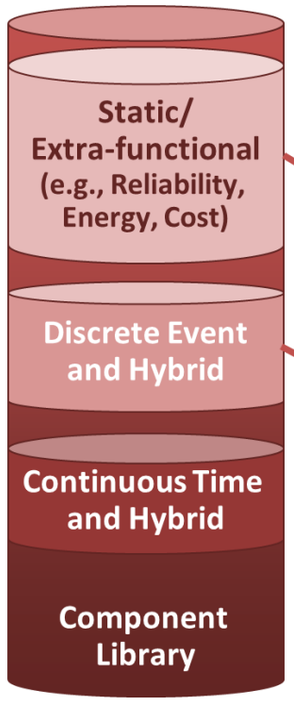


A Rigorous Calculus for Modular and Hierarchical Design



- Modular verification of "global" properties of systems out of local properties of components
- Step-wise refinement of large, complex architectures
- Design reuse

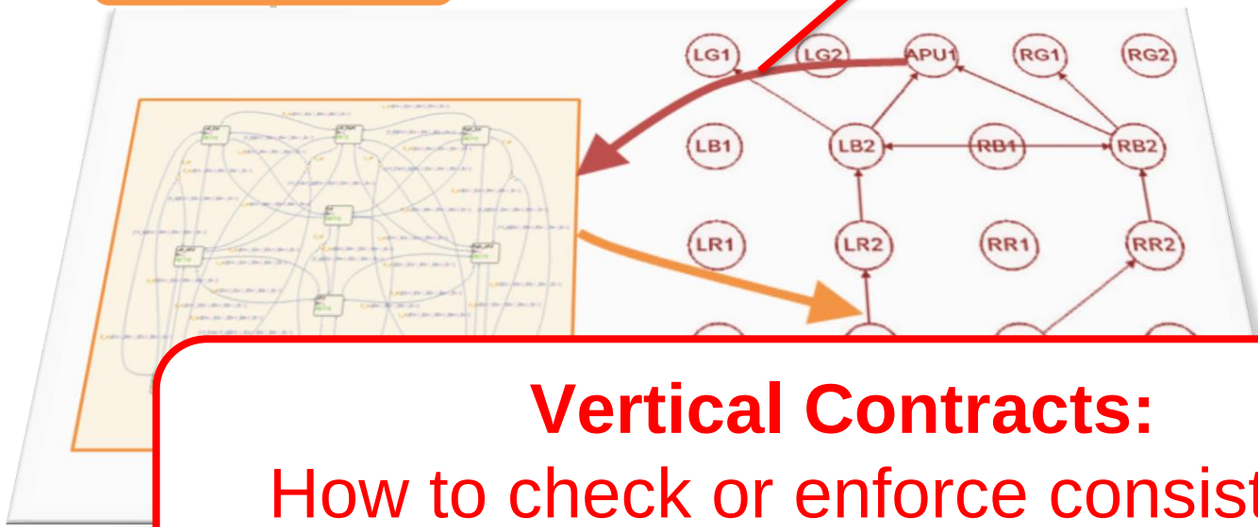
Vertical Contracts



Top-level Specification

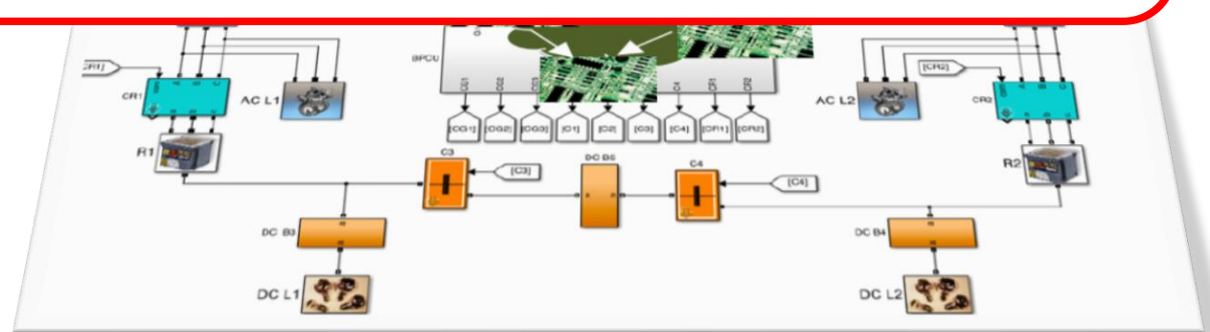
Architecture Design

Control &
Component Design



Horizontal Contracts:
How to check or enforce compatibility?

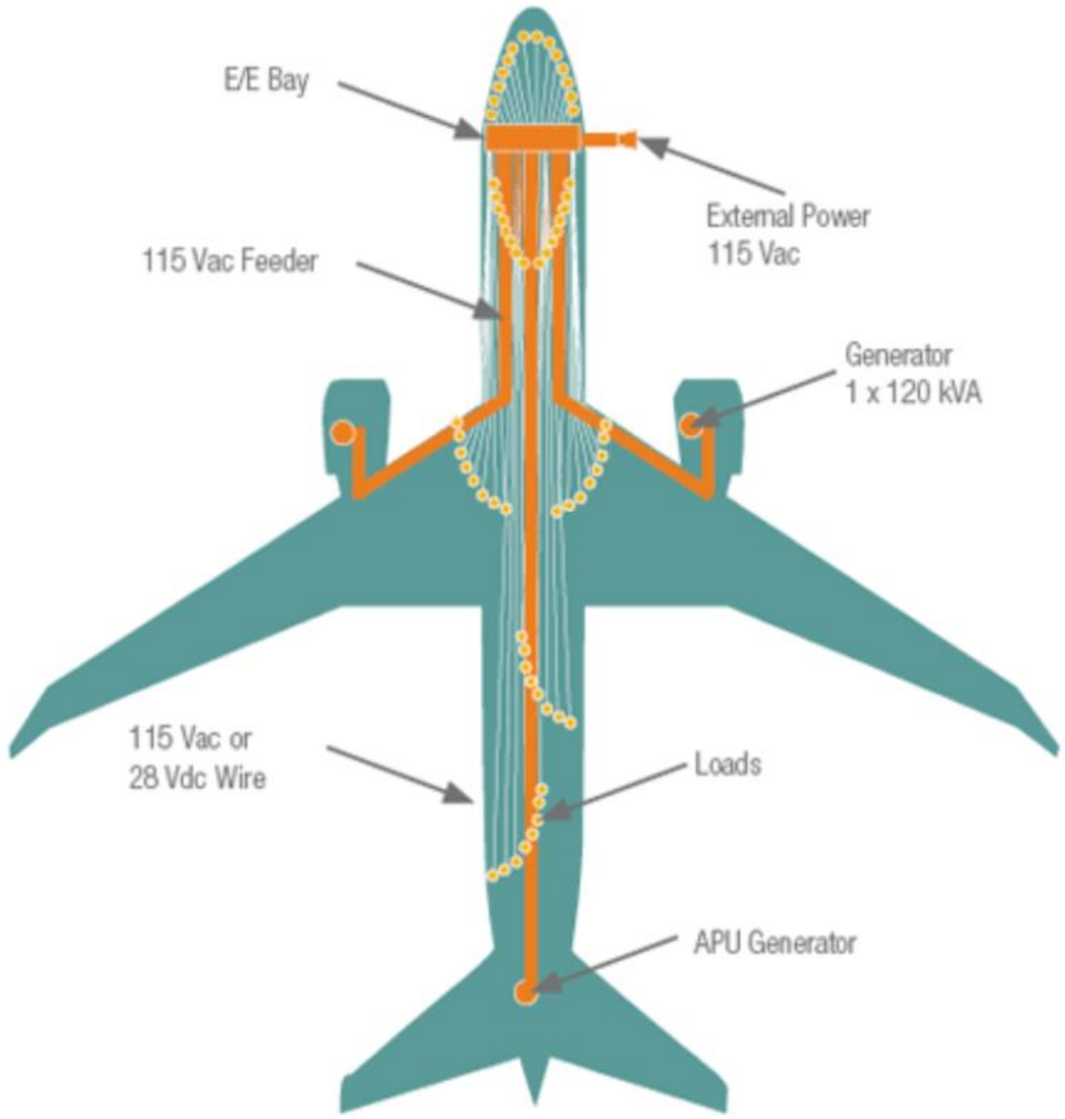
Vertical Contracts:
How to check or enforce consistency between the two levels?



*Think about the role of **design rules in physical design***

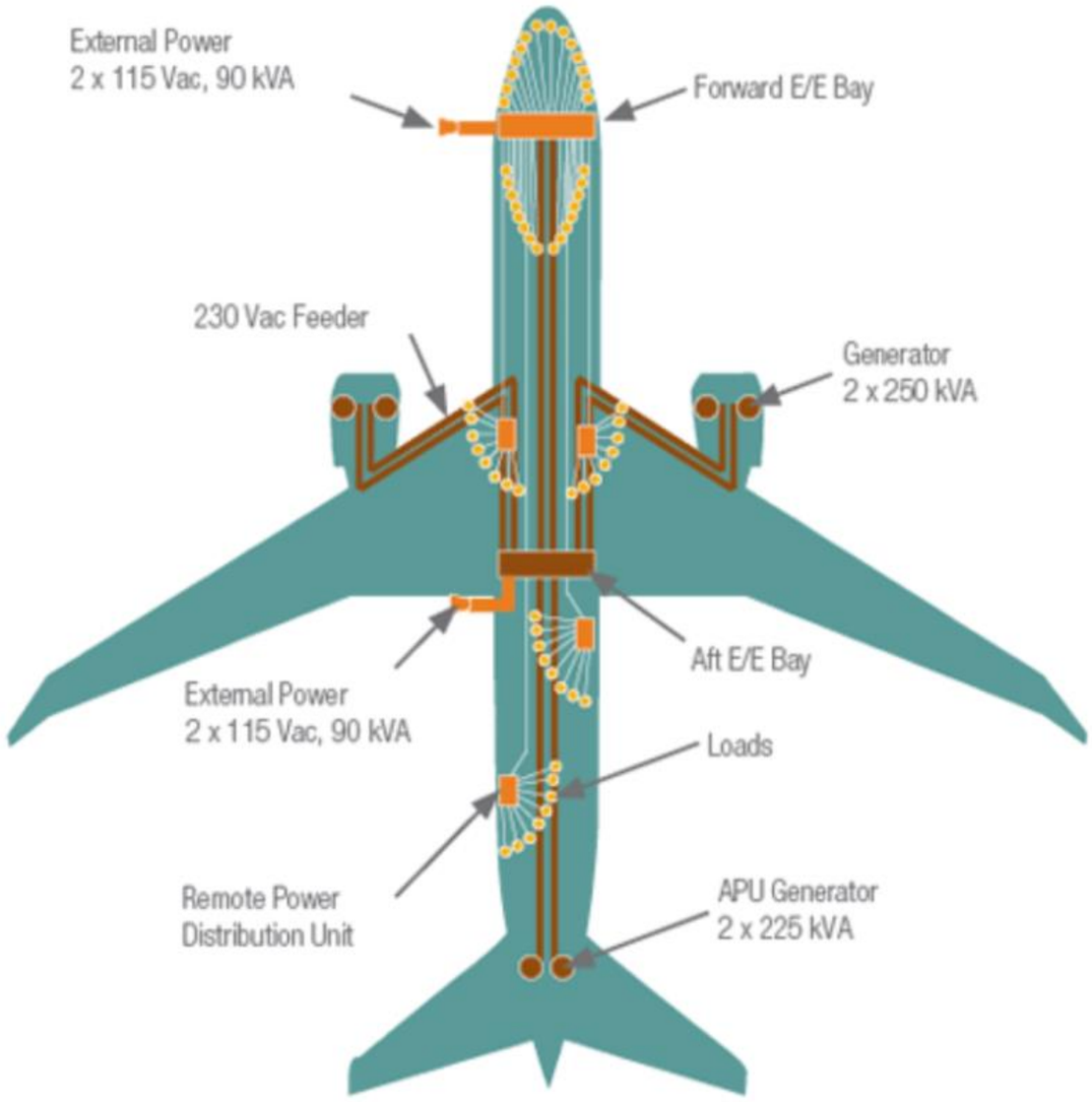
Electric Power System (EPS) in "More-Electric" Aircraft

TRADITIONAL



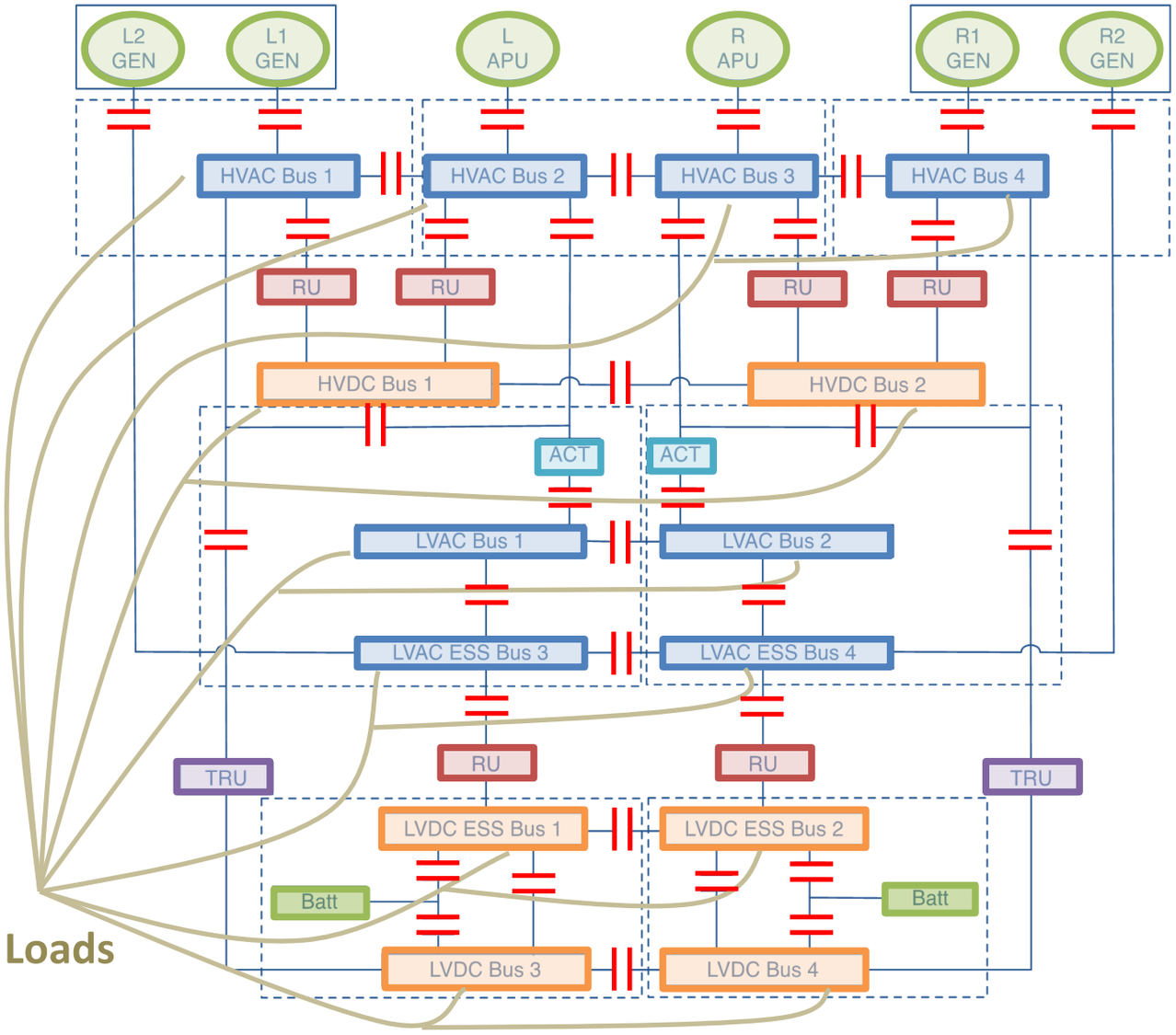
Centralized Distribution:
Circuit Breakers, Relays,
and Contactors

787



Remote Distribution:
Solid-State Power Controllers
and Contactors

Aircraft Electric Power System Design



Single Line Diagram modified from Honeywell Patent

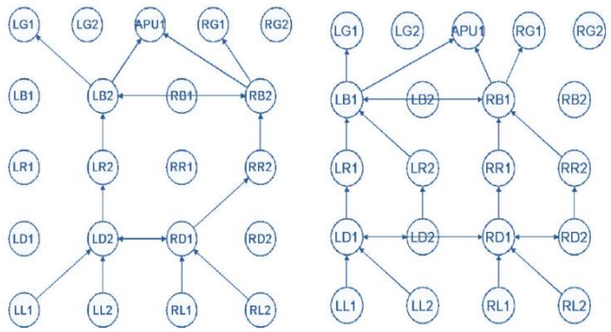
- Design architecture, i.e., the set of
  - **Generators**
  - **Batteries**
  - **AC Buses**
  - **DC Buses**
  - **Rectifiers**
  - **Transformers**
  - **Transformers & Rectifiers**
  - **Contactors**
  - **Loads**
  - **and their interconnections**
- … and the control algorithm under safety, reliability and real-time performance requirements
- Typical requirement:
  The probability that a critical bus is unpowered for more than 70 ms shall be smaller than 10^{-9}…
- **…less than 1 failure per 100,000 years of operation!**

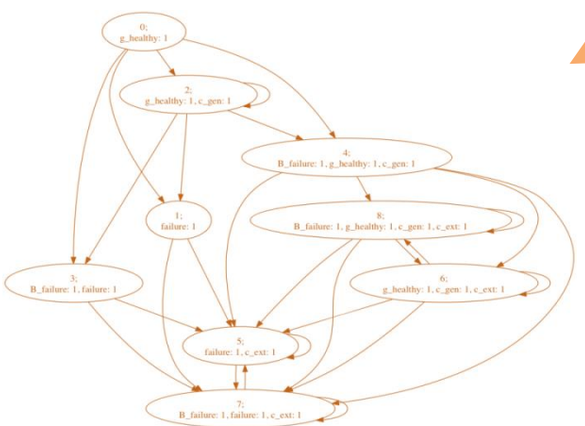
"A Contract-Based Methodology for Aircraft Electric Power System Design," IEEE Access, 2014

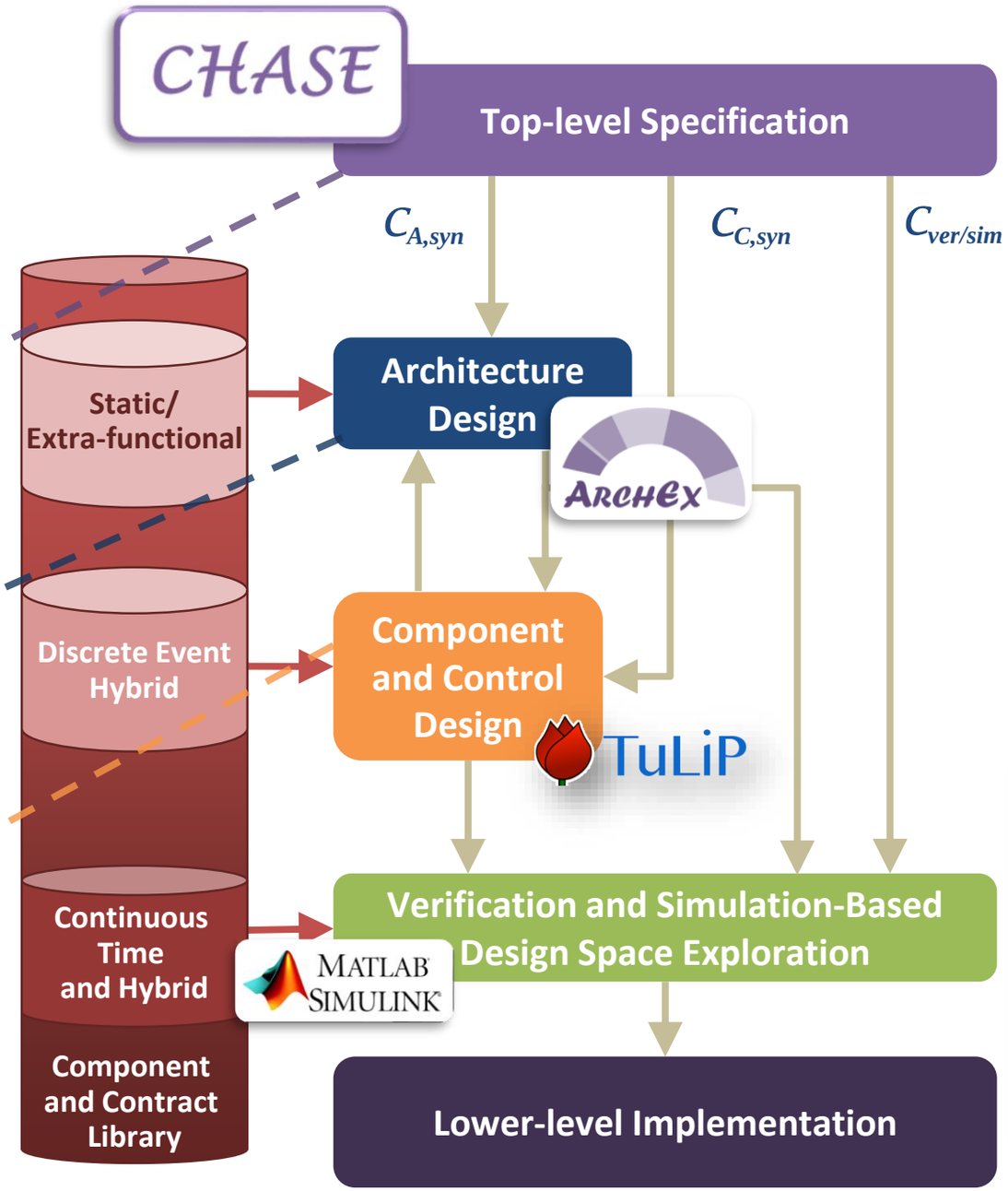
"A Platform-Based Methodology with Contracts and Related Tools for the Design of Cyber-Physcal Systems," Proc. IEEE, 2015

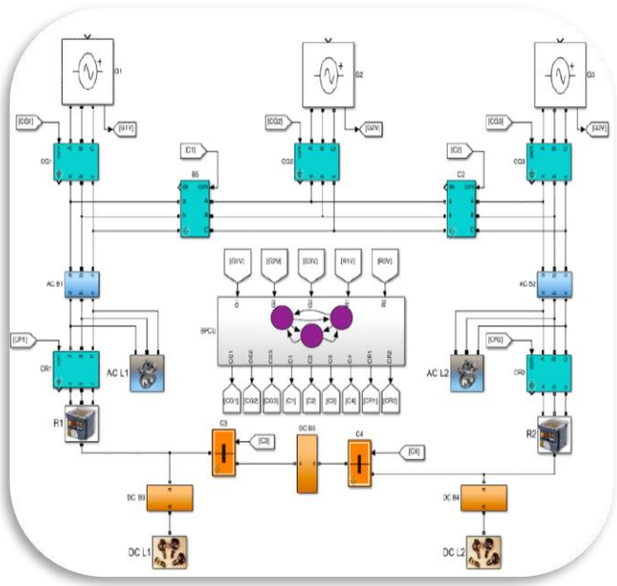
Methodology and Tools: Summary

$$\square\{(\tilde{c}=1 \wedge c=0 \wedge (x_C < T_{c_{min}})) \rightarrow (\bigcirc c=0 \wedge \bigcirc x_C = x_C + \delta)\},$$
$$\square\{(\tilde{c}=1 \wedge c=0 \wedge (x_C \geq T_{c_{min}})) \rightarrow (\bigcirc c=1 \vee \bigcirc x_C = x_C + \delta)\},$$
$$\sum_{i=1}^{n_{rec}} M^{rd}_{i,j} \geq \sum_{i=1}^{n_{load}} M^{dl}_{j,i}, \quad \sum_{i=1}^{n_{rec}} M^{rd}_{i,j} \geq \sum_{i=1}^{n_{dcb}} M^{dd}_{j,i}$$
$$\square_{[\tau_i,\infty)}(\diamondsuit_{[0,t_{max}]}(\mid V_{DC}(t) - V_d \mid < \epsilon))$$









1. No AC bus shall be simultaneously powered by more than one AC source.
2. The aircraft electric power system shall provide power with the following characteristics: 115 +/- 5 V (amplitude) and 400 Hz (frequency) for AC loads and 28 +/-2 V for DC loads.
3. The failure probability at an essential load must be less than 10^{-9} during a mission.
4. DC buses shall not be unpowered for more than 70 ms.

"Methodology and Tools for Next Generation Cyber-Physical Systems: The iCyPhy Approach," P. Nuzzo, A. Sangiovanni-Vincentelli, R. Murray, *INCOSE* 2015

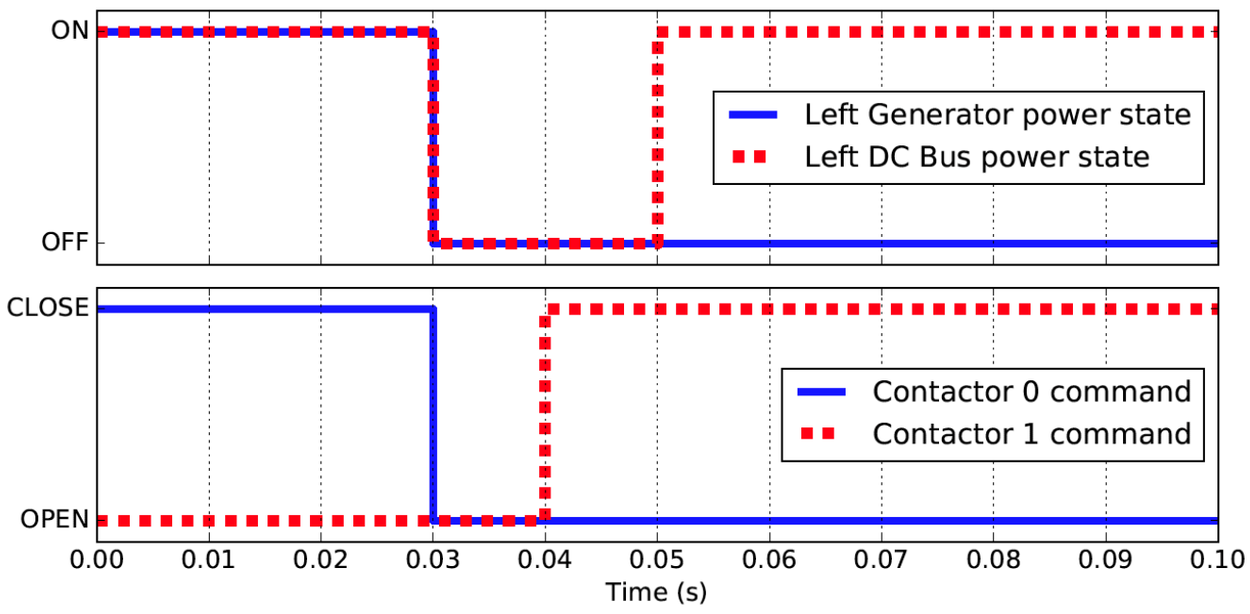
Aircraft Power System Design with CHASE

CHASE

```
1  REQUIREMENTS
2  If possible, left DC buses shall be powered by left generators.
3  If the left generator fails, the left AC bus shall be connected to another generator.
4  Failed generators must be disconnected in 20 ms or less.
5  Generators shall never be connected in parallel through AC buses.
6  When a contactor receives an open signal, it shall become open in 10 ms or less.
7  When a contactor receives a close signal, it shall become closed in 10 ms or less.
8  A DC bus shall never be disconnected from a generator for more than 30 ms.
9
10 ASSUMPTIONS
11 Generators do not recover from failures.
12 AC buses do not fail.
13 Rectifiers do not fail.
14 DC buses do not fail.
15 Loads do not fail.
16 At most 1 generator may fail.
```

Logic specification are up to 4,500 literals in size

Inconsistent when time is less than 20 ms

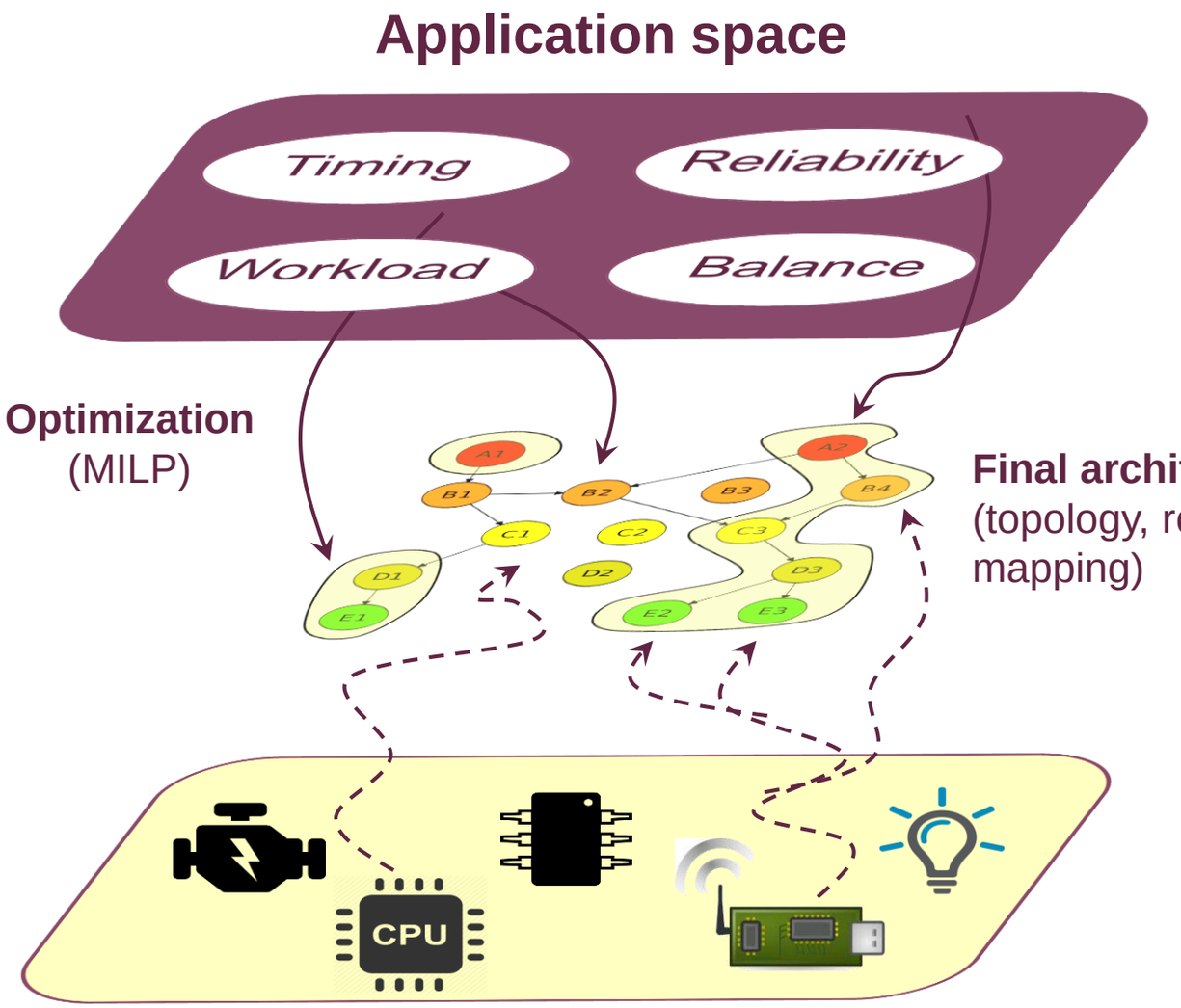



```
1  REQUIREMENTS
2  prefer-active-connection(left DC bus, left generator);
3  must-disconnect-failed(generator, 20, MS);
4  never-connect(generator, generator, AC bus);
5  always-active-connection(DC bus, generator, 30, MS);
6
7  ASSUMPTIONS
8  no-recovery(generator);
9  no-failures(AC bus);
10 no-failures(Rectifier);
11 no-failures(DC bus);
12 no-failures(load);
13 switch-on-time(contactor, 10, MS);
14 switch-off-time(contactor, 10, MS);
15 max-failures(generator, 1);
```

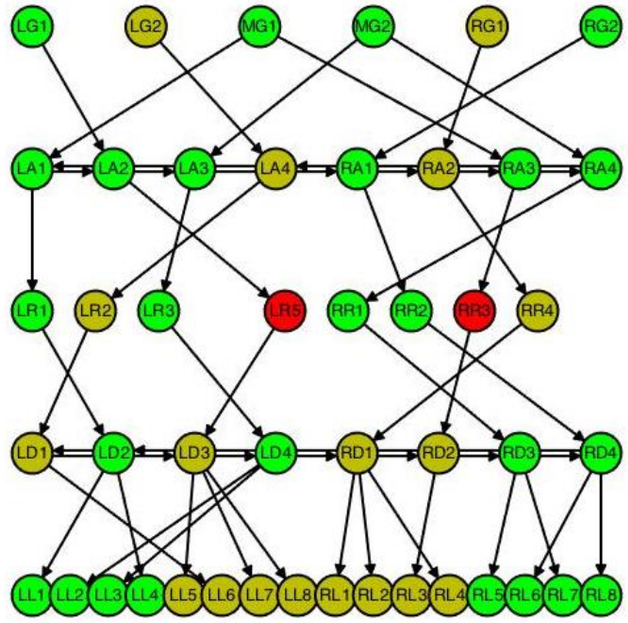
"CHASE: Contract-Based Requirement Engineering for Cyber-Physical System Design," P. Nuzzo et al., DATE, 2018

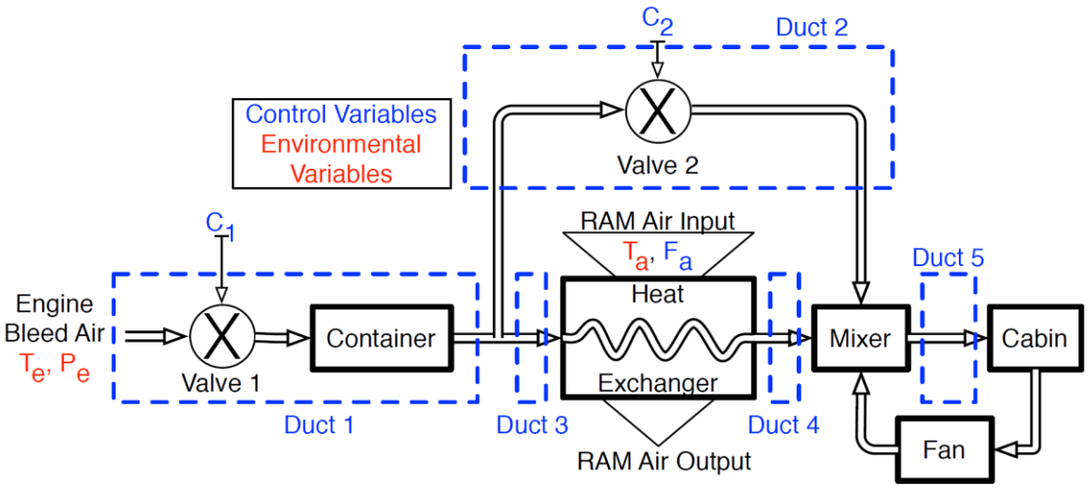
Demonstrated reasoning about temporal properties of networks and integration with Natural Language Processing tools (IBM WATSON)

Optimized Selection of Reliable and Cost-Effective Architectures



Dreamliner-like power system based on Honeywell patent reproduced in ~4 min



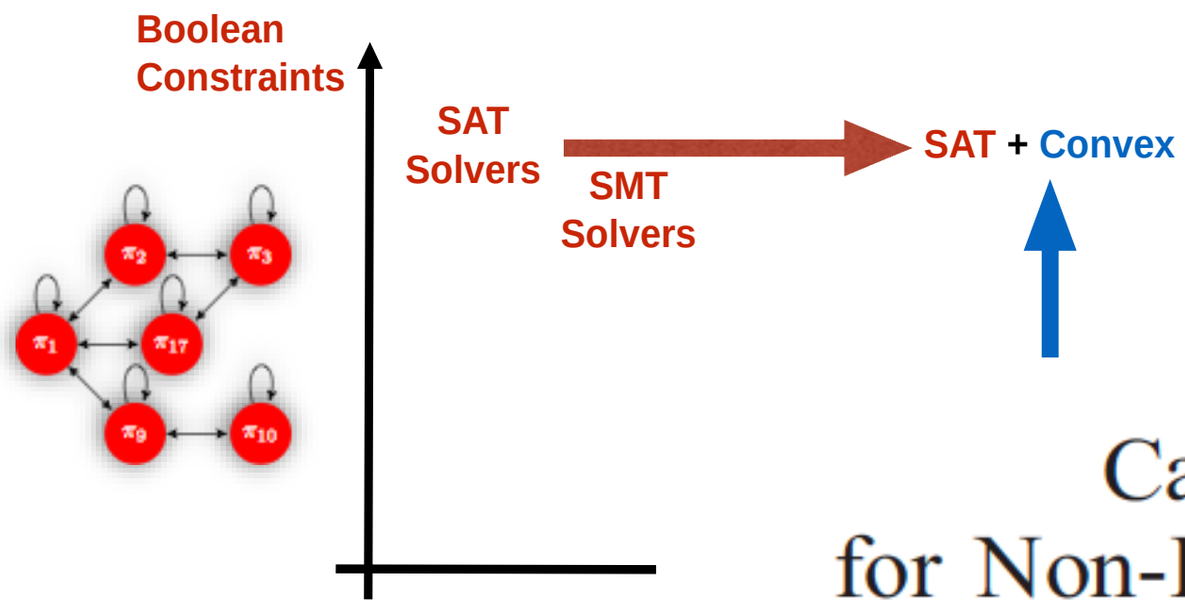


*Architecture exploration of **aircraft air management** systems*

"Optimized Selection of Reliable and Cost-Effective Cyber-Physical System Architectures," ***DATE'14***

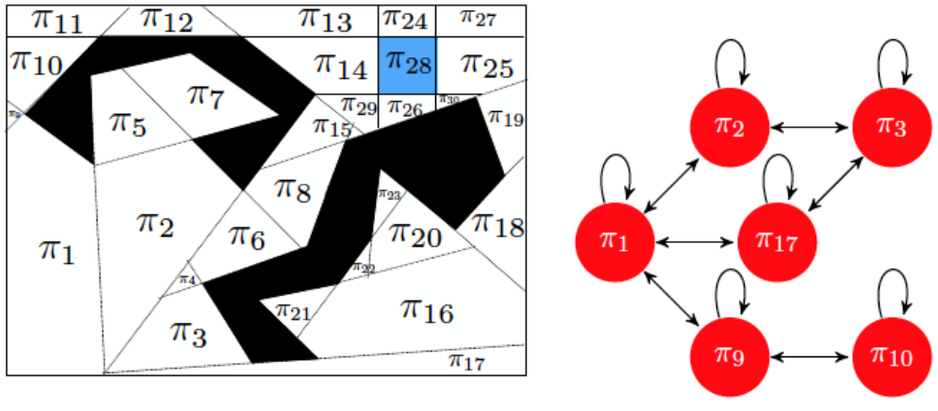
"A Mixed Discrete-Continuous Optimization Scheme for Cyber-Physical System Architecture Exploration," ***ICCAD'15***

Reasoning About Software and Dynamics: Satisfiability Modulo Convex Programming (SMC)

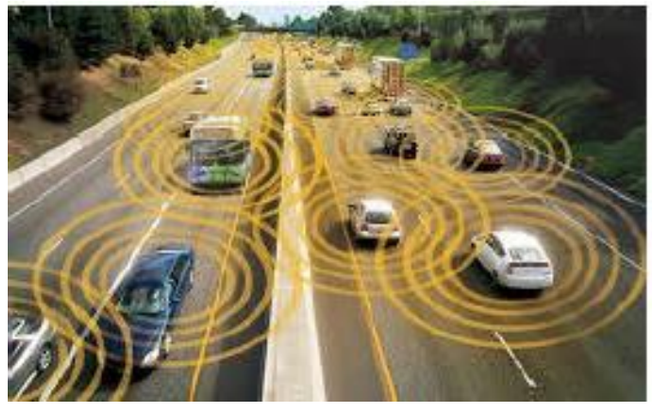


CalCS: SMT Solving for Non-Linear Convex Constraints

Pierluigi Nuzzo, Alberto Puggelli, Sanjit A. Seshia and Alberto Sangiovanni-Vincentelli
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
Berkeley, California 94720
Email: {nuzzo, puggelli, sseshia, alberto}@eecs.berkeley.edu



Controller Synthesis for Robotic Motion Planning [CDC'16, HSCC'17, CDC'17, ICRA'19]



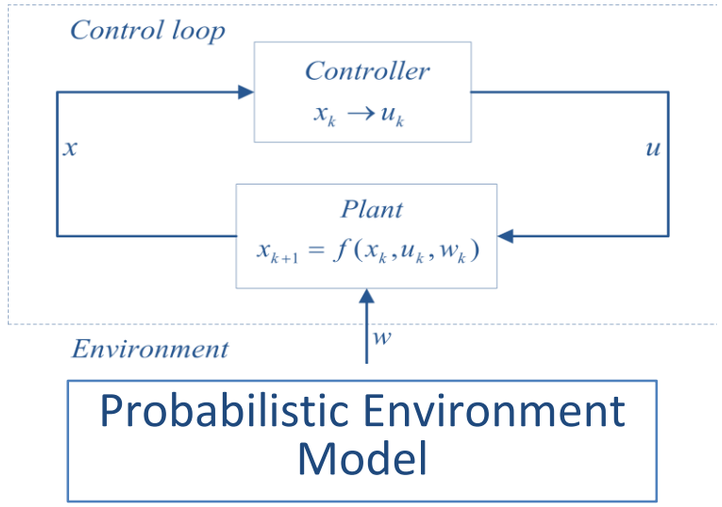
Secure State Estimation [ICCPS'16, TAC 17, TECS 18]

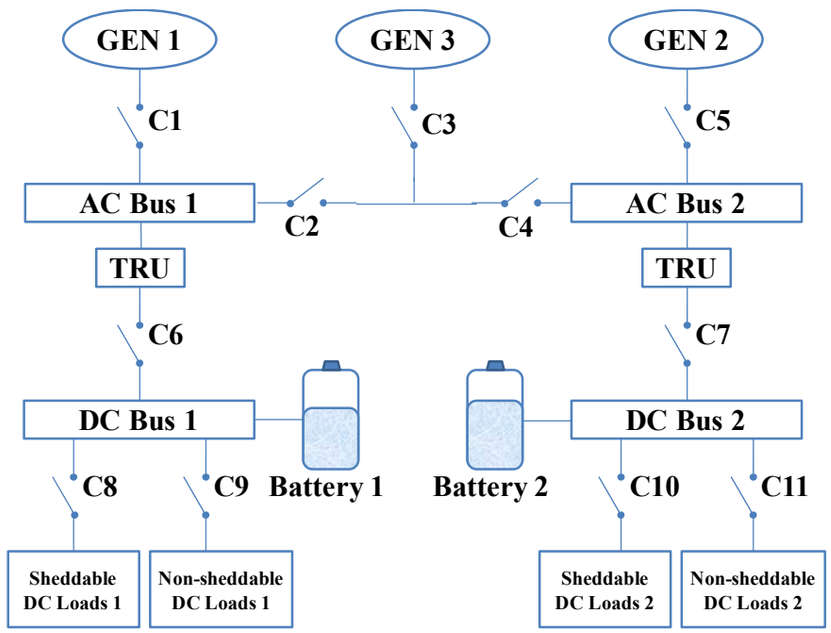
Stochastic Contracts for CPS Design with Uncertainty

Expressed in Stochastic Signal Temporal Logic (StSTL) to support probabilistic constraints

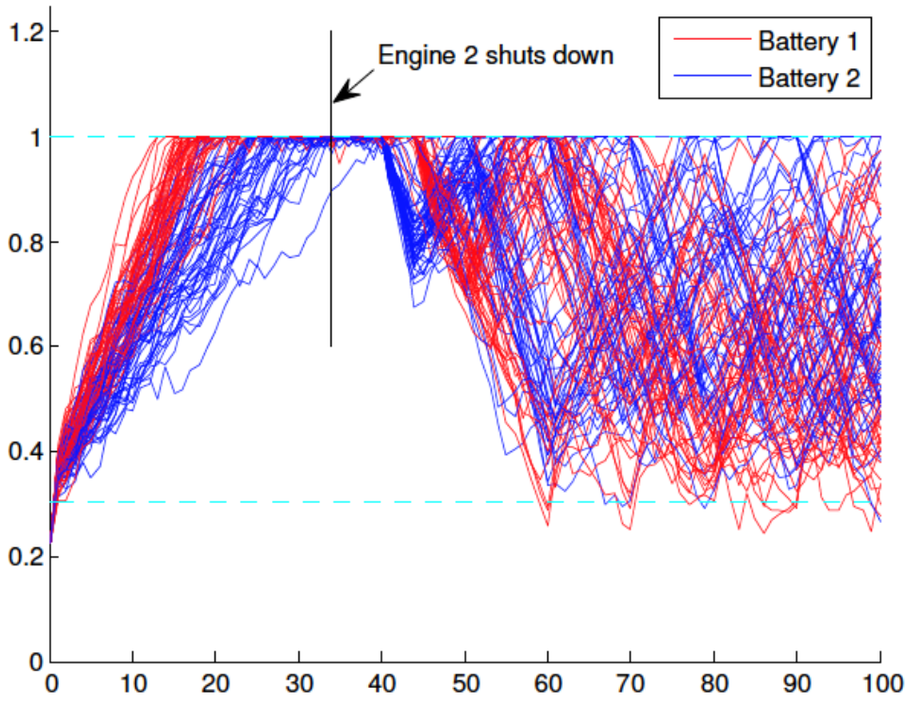
Balance expressiveness with tractability of verification and synthesis

"The battery charge level *B* shall not be less than 0.3 with probability larger than or equal to 0.95"





Stochastic Model of Aircraft Power System [TECS 19]



Battery charge versus time (50 simulations)

What's Next?

- Compositional (modular, hierarchical) abstractions for CPS design

- Computational tools for reasoning about the interaction between discrete and continuous models

- Dealing with uncertainty

USC Viterbi
School of Engineering
Center for Cyber-Physical Systems and the Internet of Things

Thank you