A Formal Contract-Based Design Methodology for CyberPhysical Systems

Alberto Sangiovanni Vincentelli

The Edgar L. and Harold H. Buttner Chair

Department of Electrical Engineering and Computer Sciences

University of California at Berkeley

Cofounder and Member of the Board,

Cadence Design Systems, San Jose', California

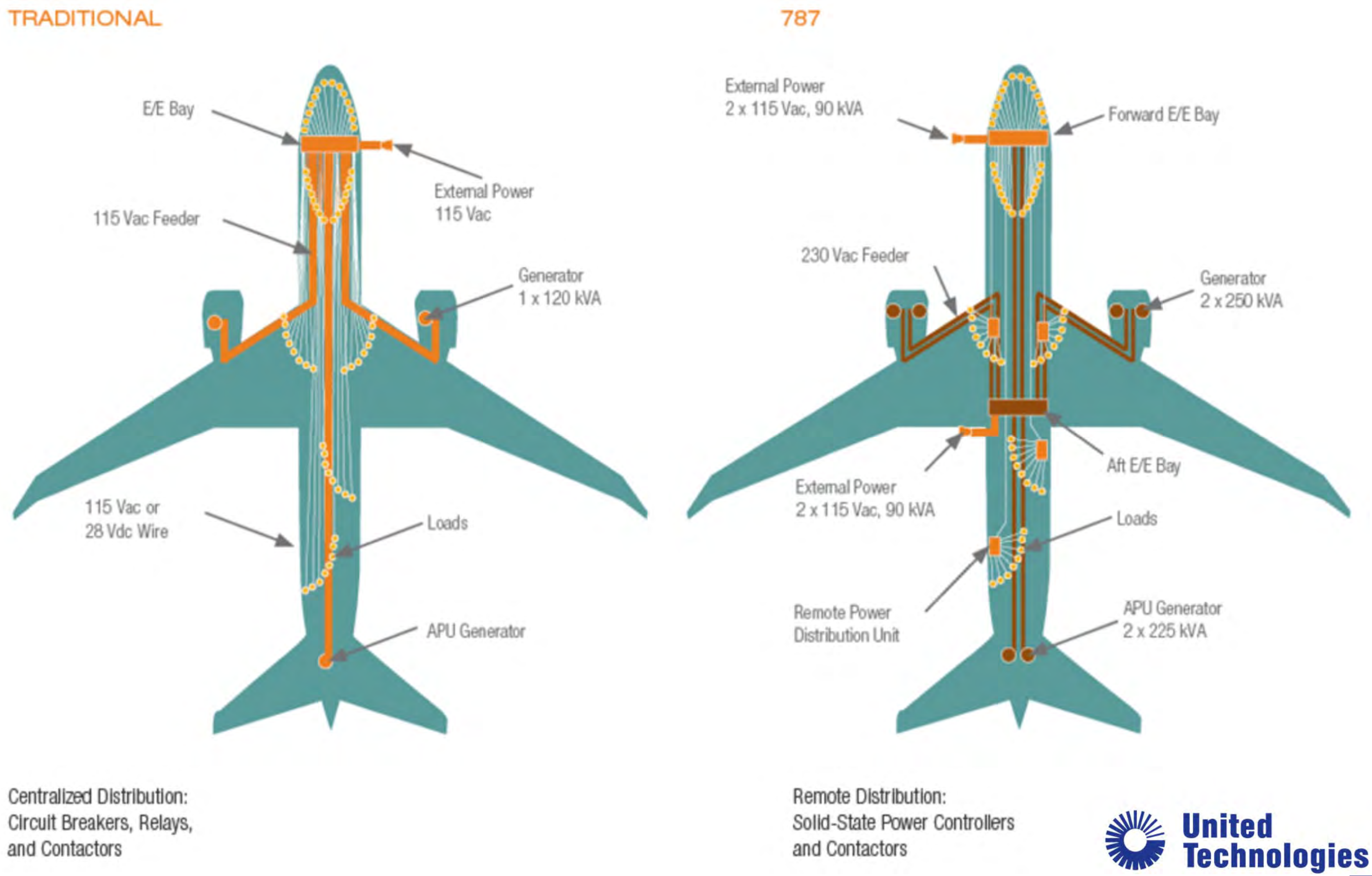
iCyPhy

TerraSwarm

Acknowledgement

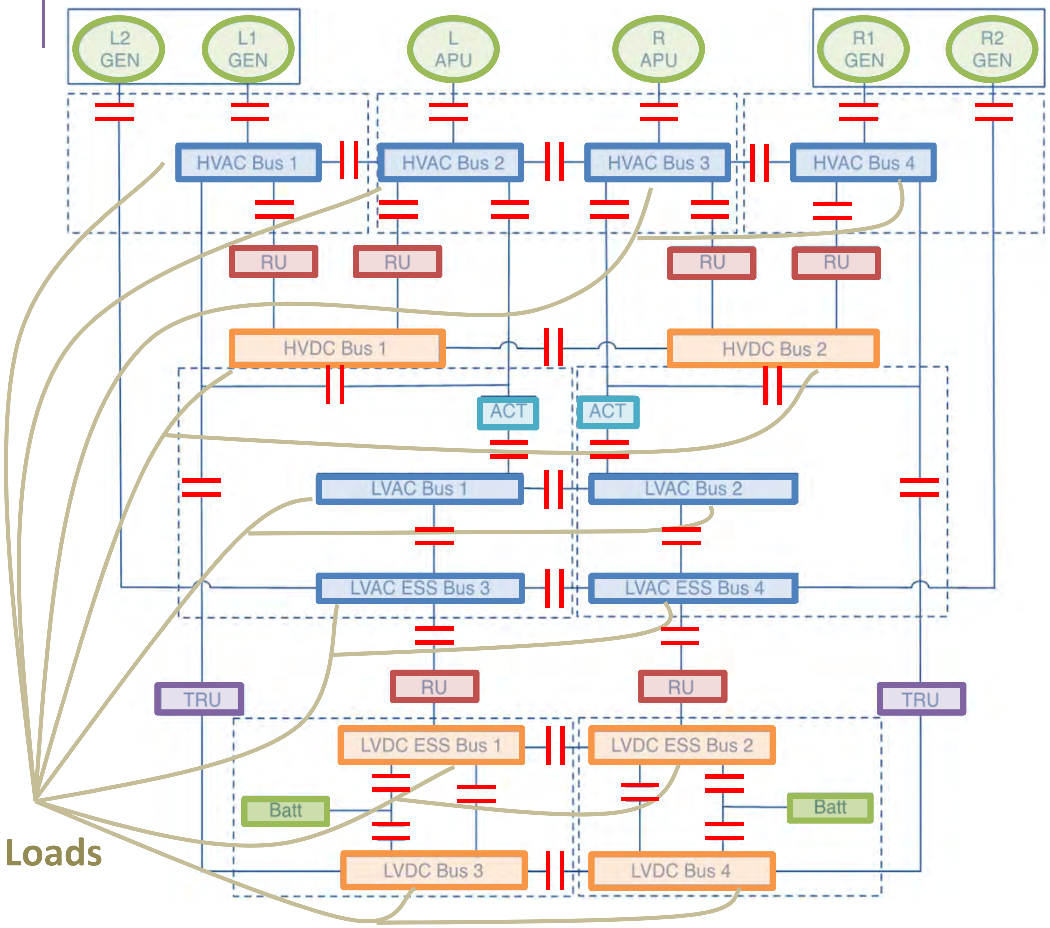
- Slides in collaboration with Pierluigi Nuzzo
- Research collaboration: P. Nuzzo, R. Passerone, A. Benveniste, W. Damm, A. Iannopollo, I. Incer

Running Example: Electric Power System (EPS) in “More-Electric” Aircraft



United Technologies

Running Example:
Aircraft Electric Power System Design

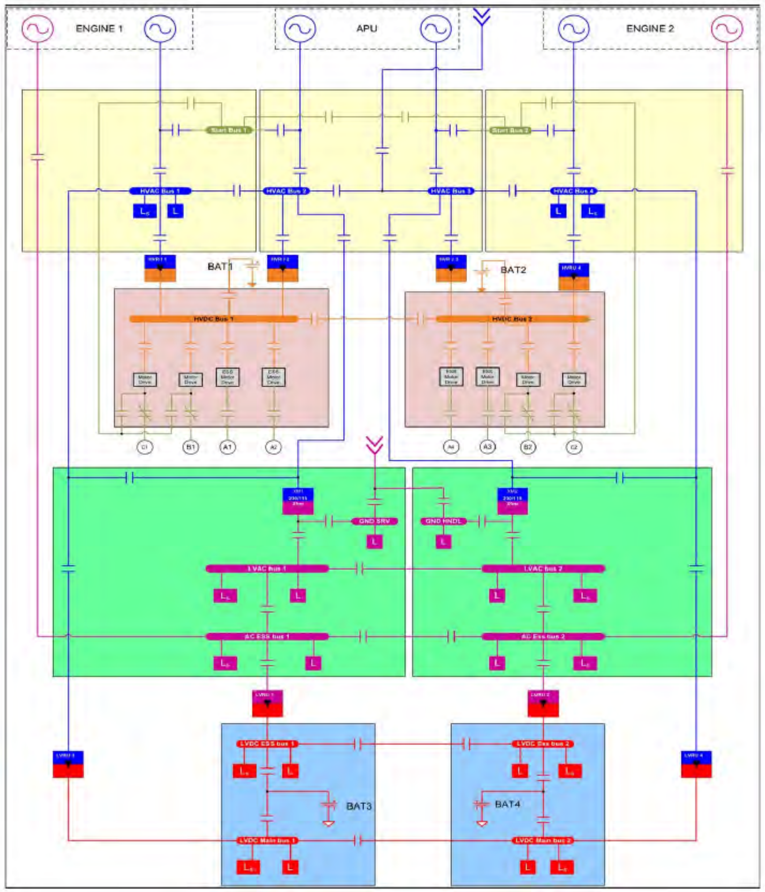


Single Line Diagram modified from Honeywell Patent

- 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}

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

Running Example: Electric Power System (EPS) in "More-Electric" Aircraft



Single Line Diagram modified from Honeywell Patent

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 and 400 Hz for AC loads and 28 +/-2 V for DC loads.
3. The **failure probability** of a critical DC bus must be less than 10^{-9} during a mission.
4. Critical DC buses shall not be unpowered for more than 50 ms.
5. …

State-of-The-Art: The V-Model

*Conventional methodologies can lead to inefficient or **incorrect** implementations, long re-design cycles, **cost** overruns, unacceptable **delays***


THE WALL STREET JOURNAL.

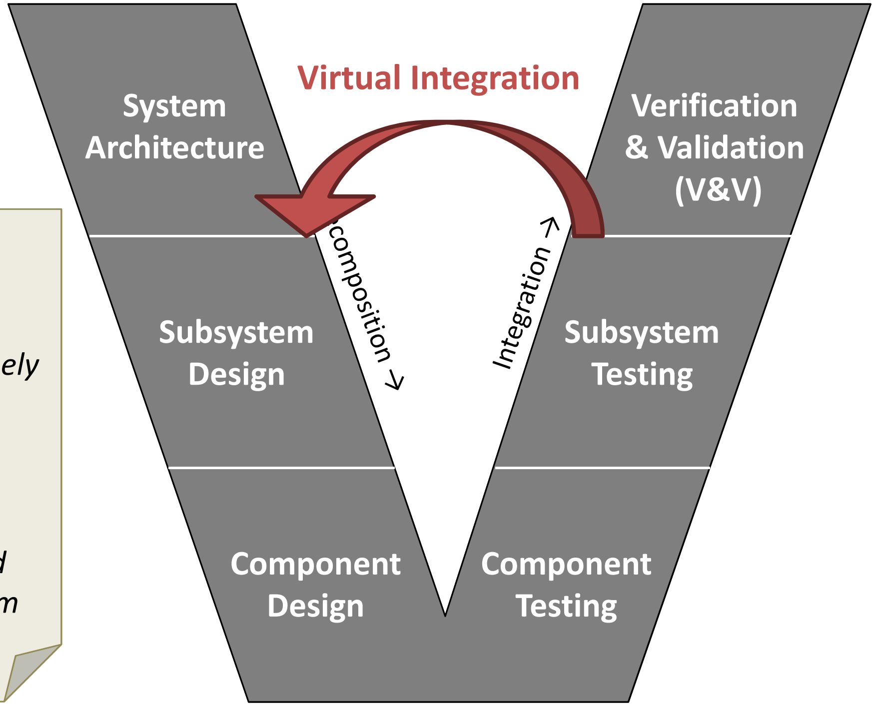
March 26, 2012, 4:45 PM

BMW Recalling 1.3 Million Cars To Fix Electrical Flaw

March 26, 2012

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

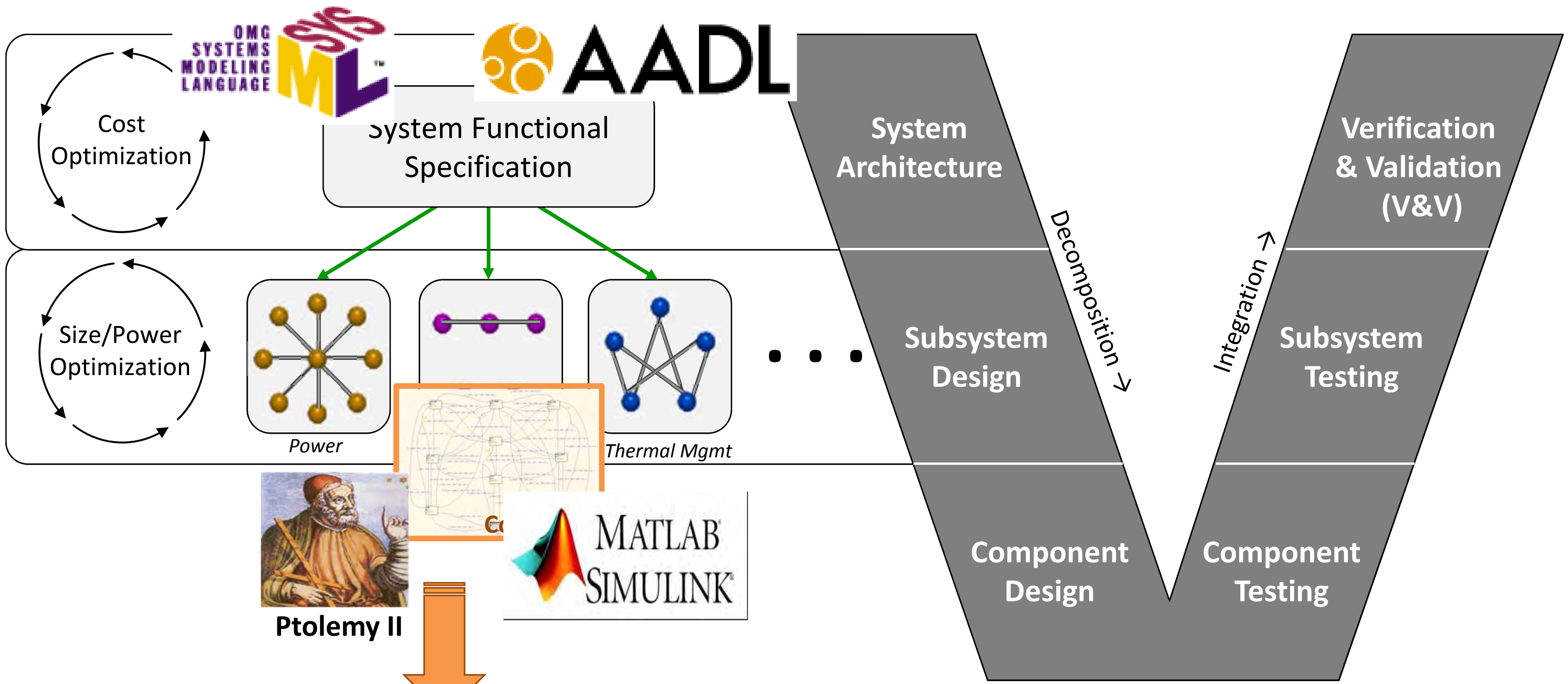
*Design process **arbitrarily** decomposes system and largely ignores **complexity** — undesired and multi-mode interactions and emergent system behaviors*

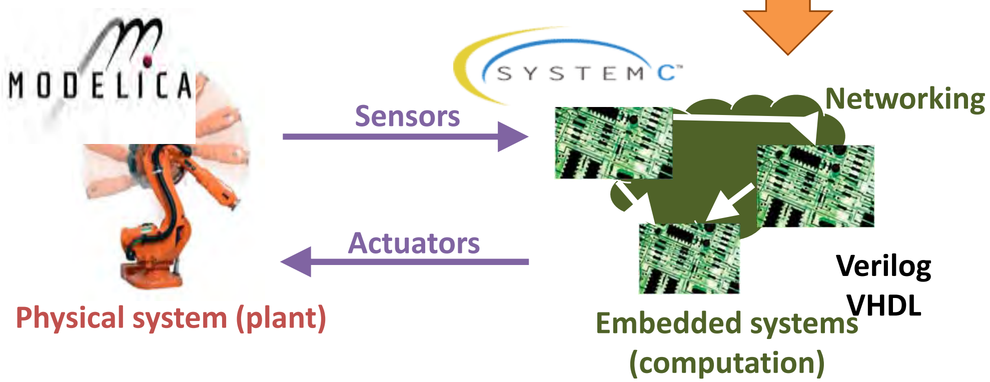


*Need more support for scalable **design space exploration**, early detection of **requirement inconsistencies,** more **scalable** verification and validation methods*

[Nuzzo and ASV, “Let’s Get Physical: Computer Science Meets Systems”, FPS’14]

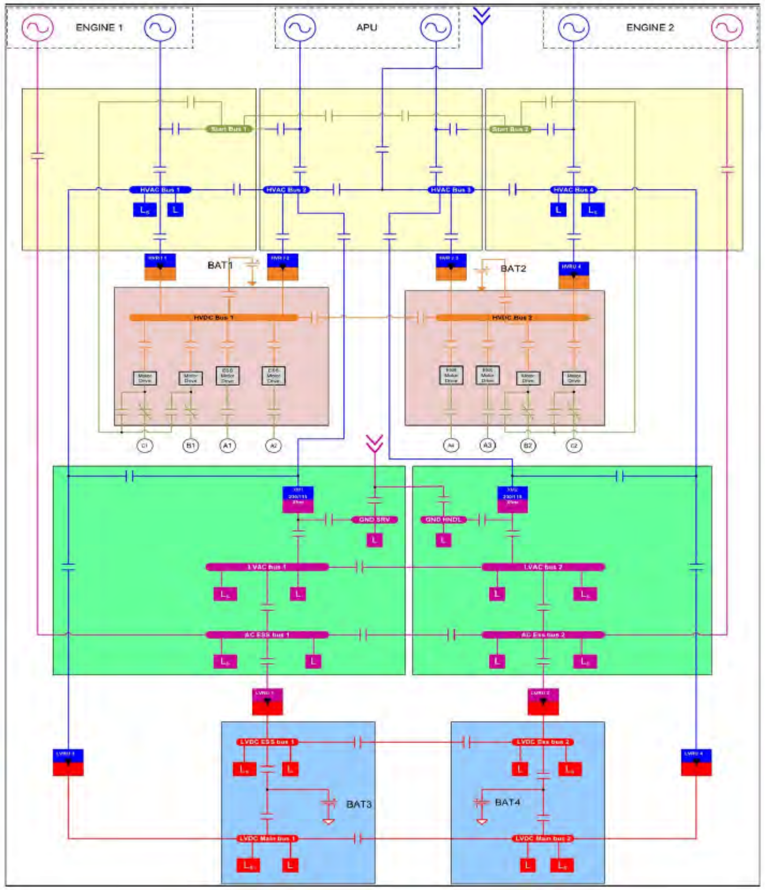
State-of-The-Art: Tools





*Need a **rigorous** framework that can **integrate** existing design techniques and tools from **multiple** disciplines to perform **design-space exploration** across different domains*

Running Example: Electric Power System (EPS) in "More-Electric" Aircraft



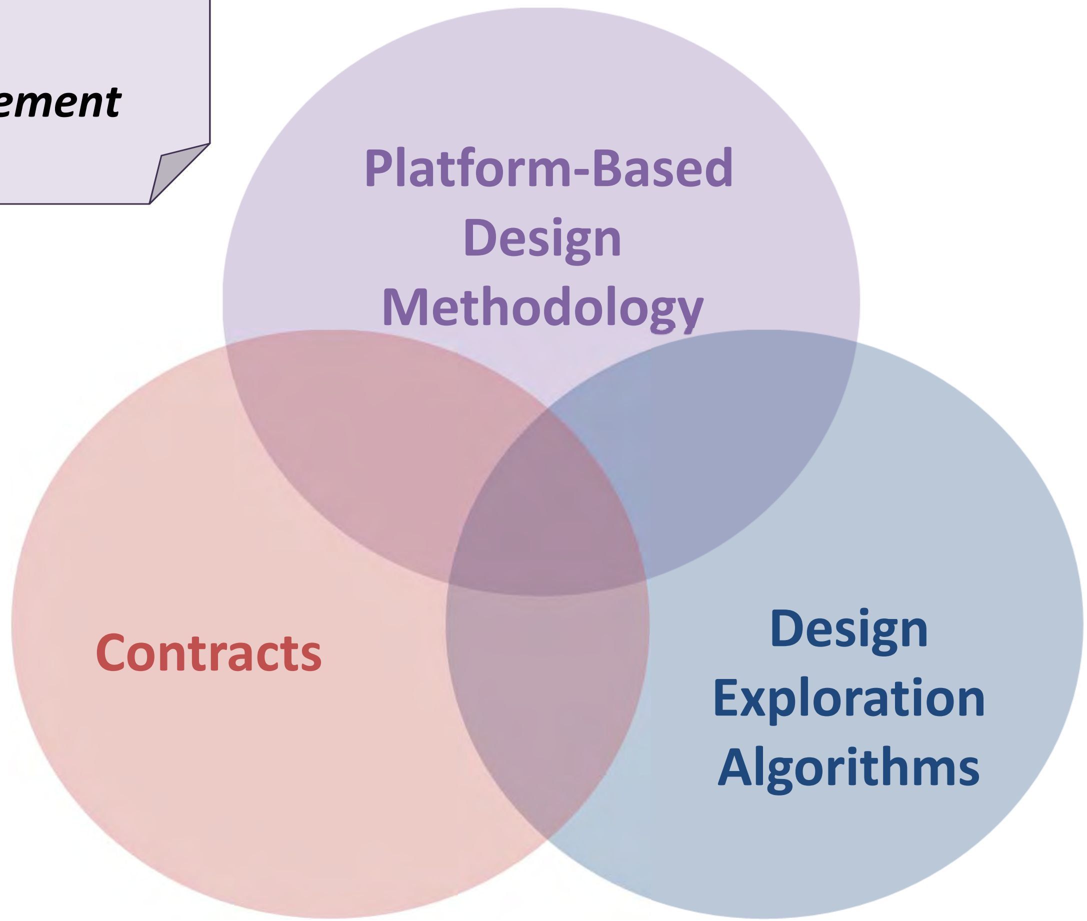
Single Line Diagram modified from Honeywell Patent

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** of a critical DC bus must be less than 10^{-9} during a mission.
4. Critical DC buses shall not be unpowered for more than 50 ms.
5. …

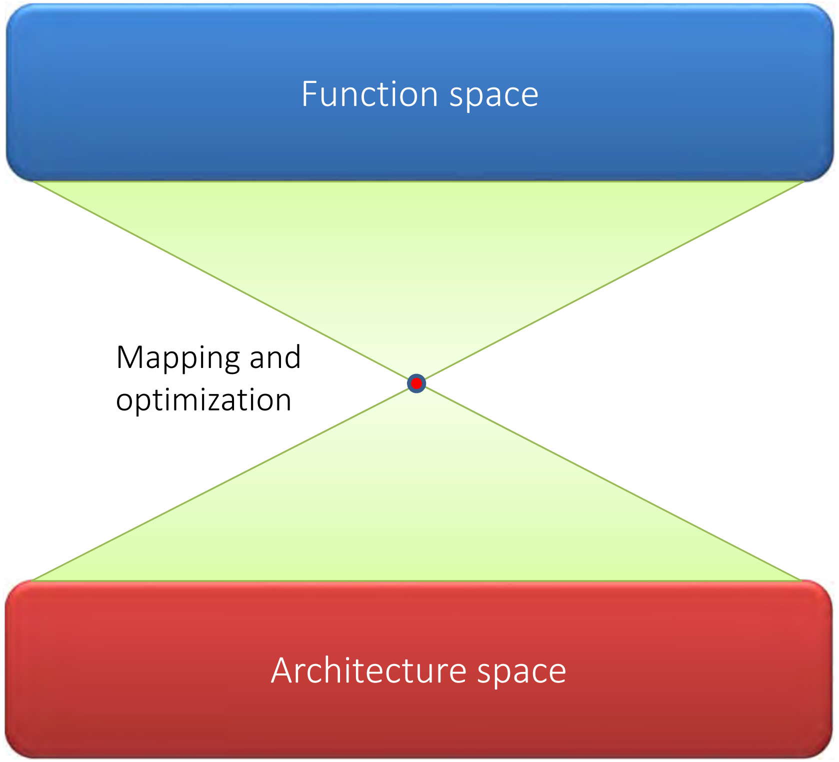
Can we address such a heterogeneous set of requirements in a hierarchical and modular way?

Addressing Cyber-Physical System Design

*Need a **comprehensive** framework that:*
- *enables **design-space exploration** across different domains in a **scalable** way*
- ***integrates** design techniques and tools from **multiple** disciplines*
- *enables early detection of **requirement inconsistencies***



Platform Based Design (PBD)

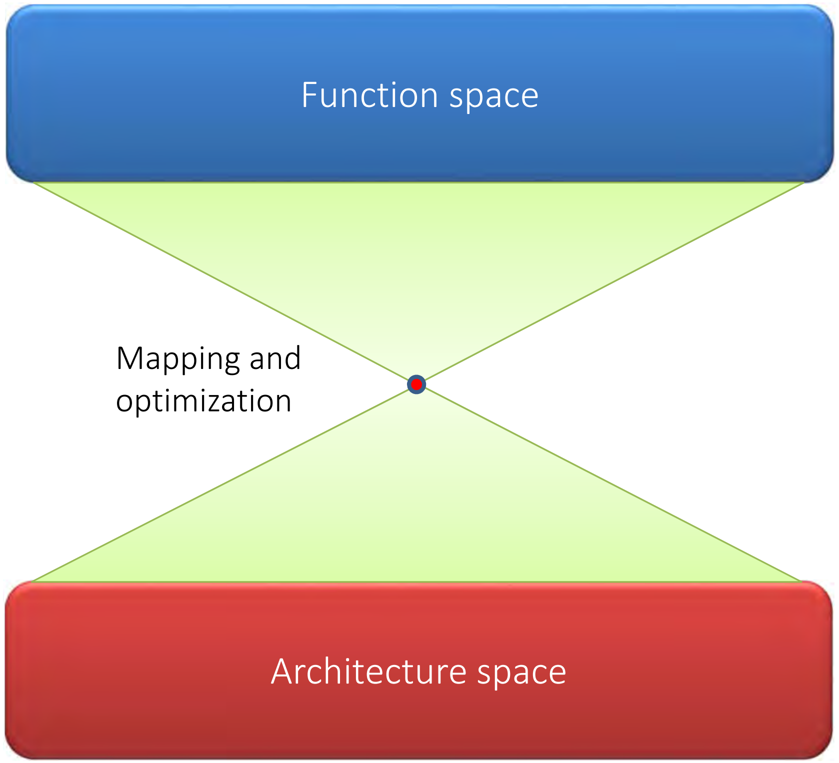


WHAT THE SYSTEM SHOULD DO.

In a layered approach, the function space includes the specification for the current mapping process. A specification can be provided by the designer or be the result of another PBD iteration.

Courtesy: Antonio Iannopollo, UCB

Platform Based Design (PBD)



WHAT THE SYSTEM SHOULD DO.

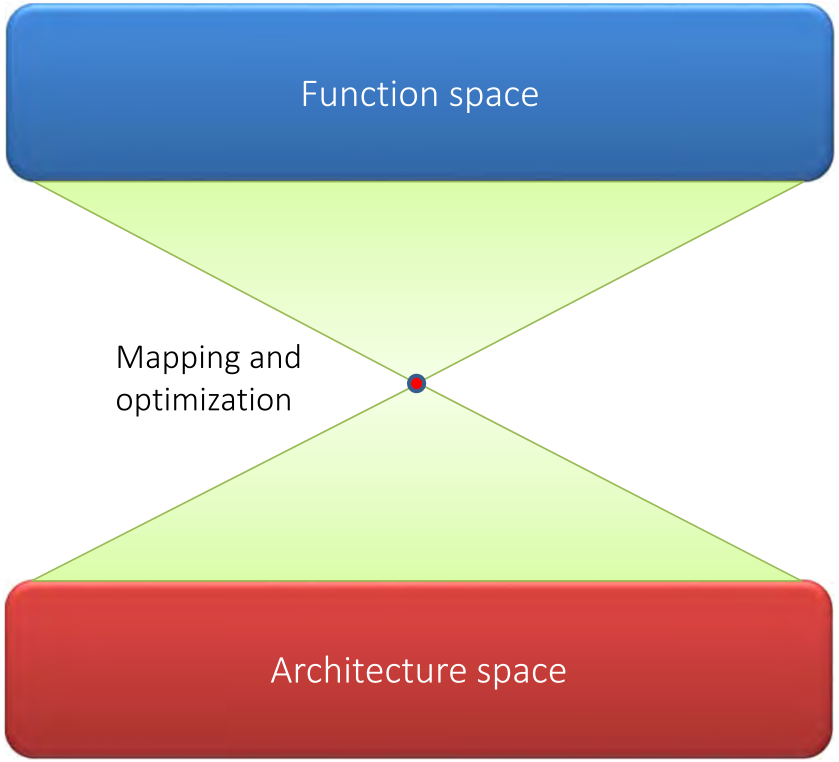
In a layered approach, the function space includes the specification for the current mapping process. A specification can be provided by the designer or be the result of another PBD iteration

HOW IT COULD BE DONE.

The architectural space includes platform components (libraries) abstracted from lower levels, connection rules and other properties such as component cost and timing properties

Courtesy: Antonio Iannopollo, UCB

Platform Based Design (PBD)



In a layered approach, the function space includes the specification for the current mapping process. A specification can be provided by the designer or be the result of another PBD iteration

The mapping process consists in the selection of a specific architectural instance, evaluating costs and functional/architectural constraints

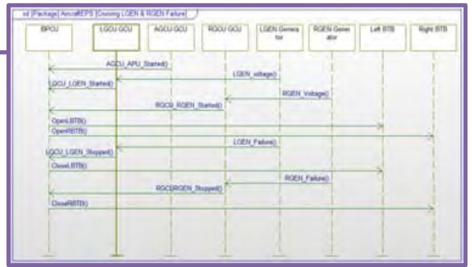
The architectural space includes platform components (libraries) abstracted from lower levels, connection rules and other properties such as component cost and timing properties

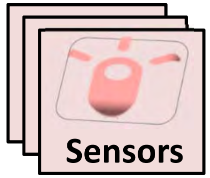
Courtesy: Antonio Iannopollo, UCB

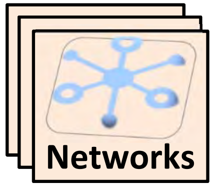
Preview of Our Strategy:
Abstract CPS Components With Contracts

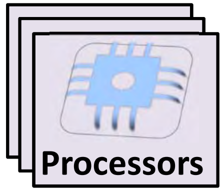
Requirements

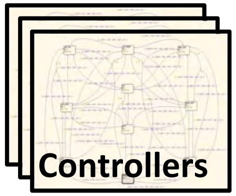
1.Reliability
2. Safety
3. Performance
4. Cost (e.g. energy, weight,…)





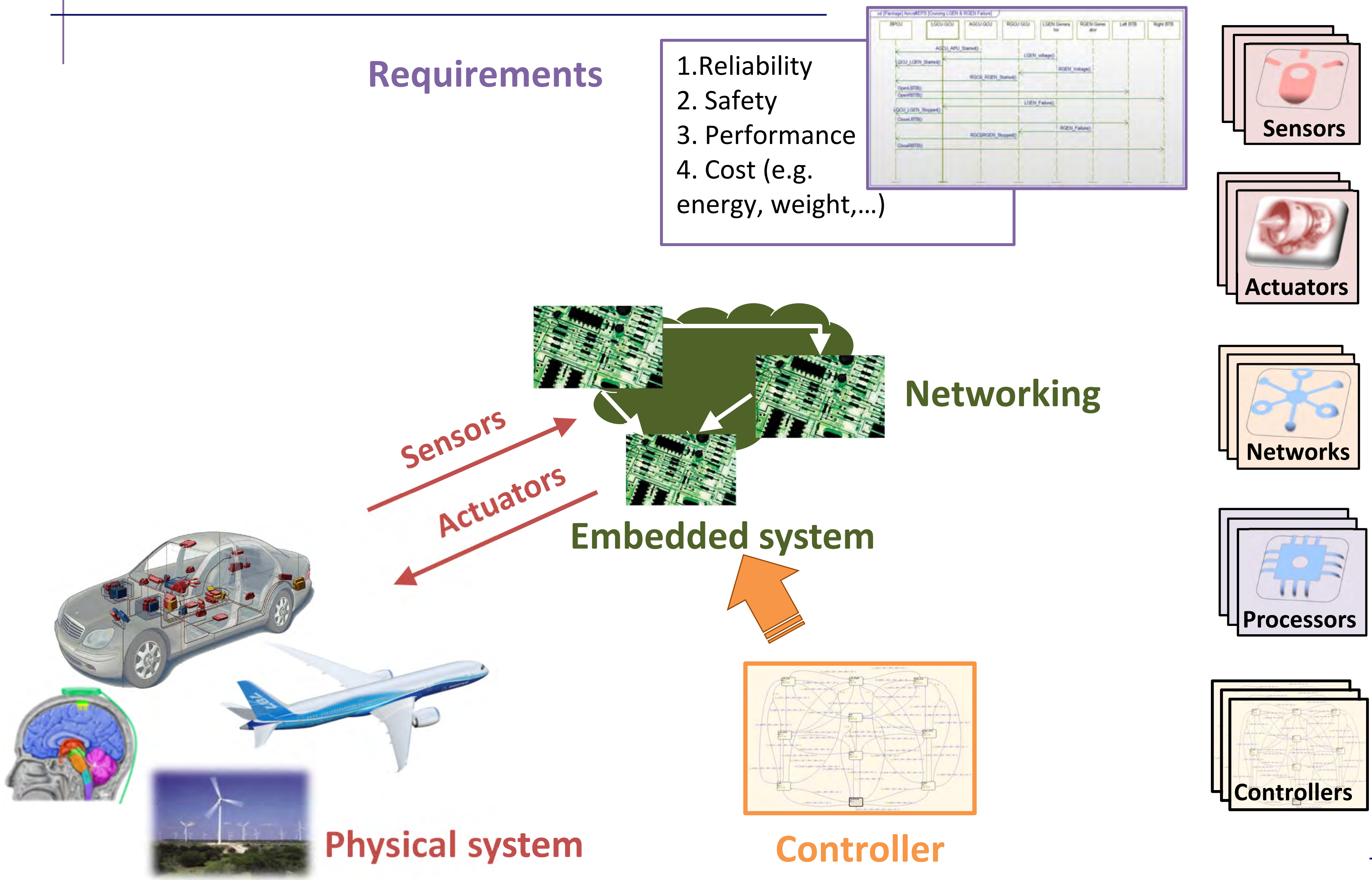






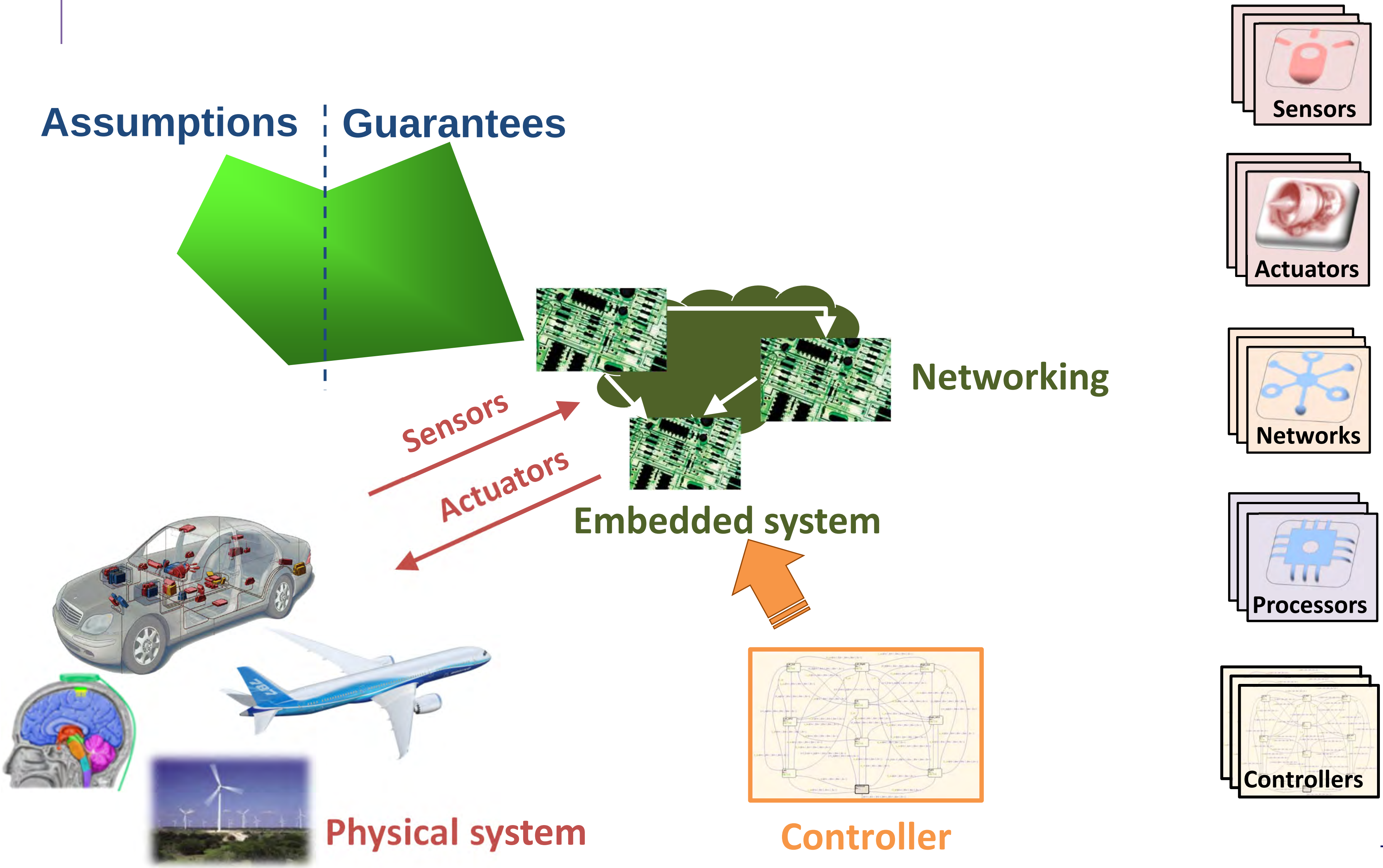
Physical system

Preview of Our Strategy:
Abstract CPS Components With Contracts

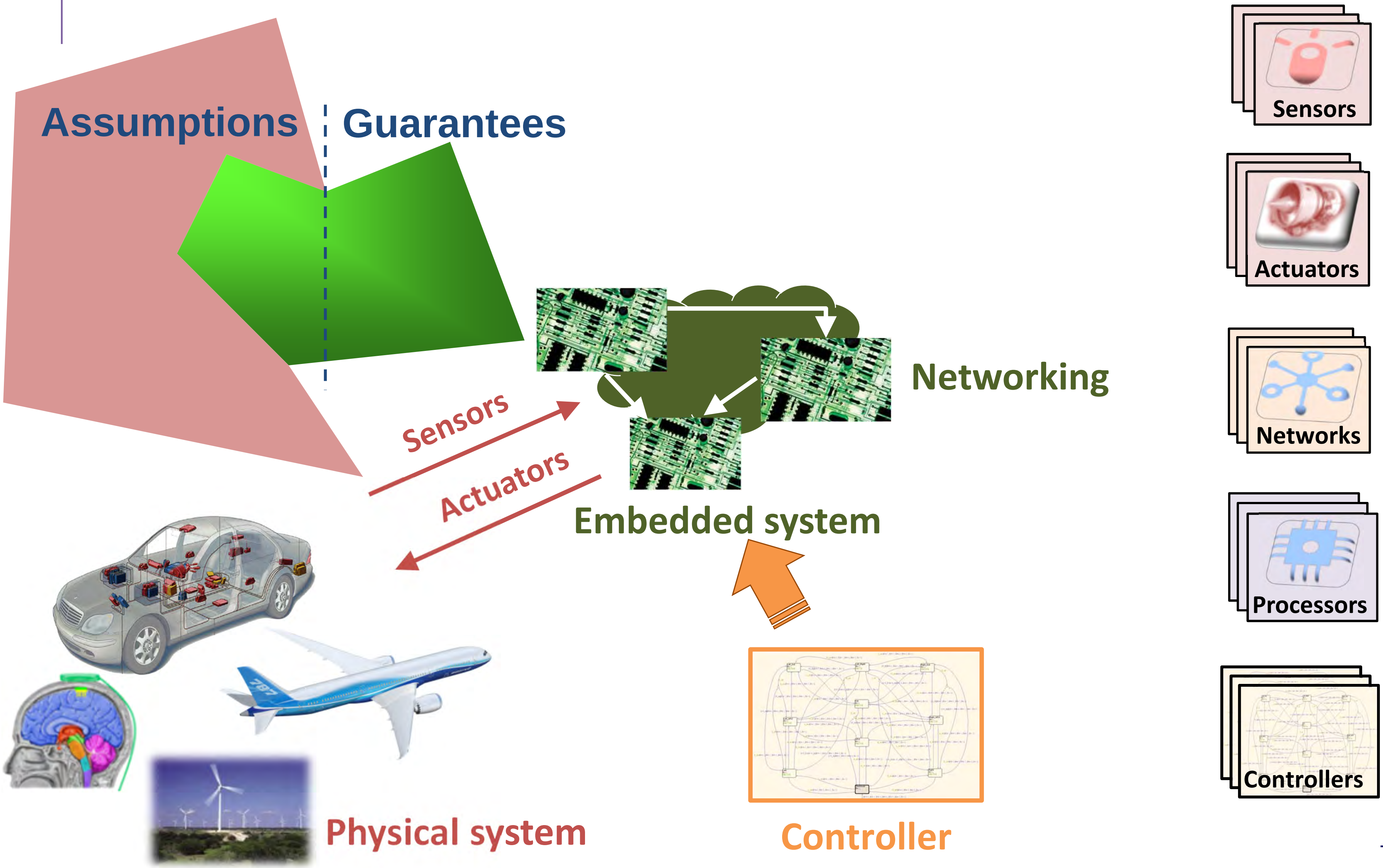


Preview of Our Strategy:

Abstract CPS Components With Contracts

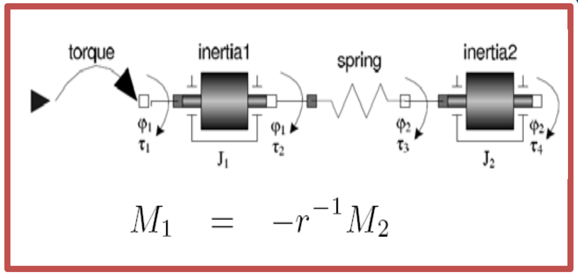


Preview of Our Strategy:
Abstract CPS Components With Contracts



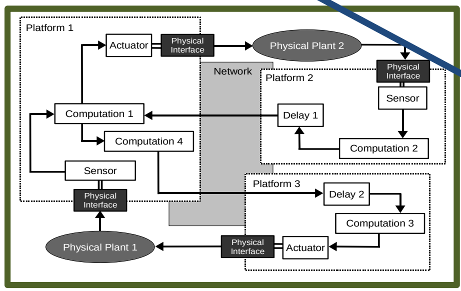
Preview of Our Strategy:
Contracts Provide Formal Support to CPS Design

Composition:
Compatible?



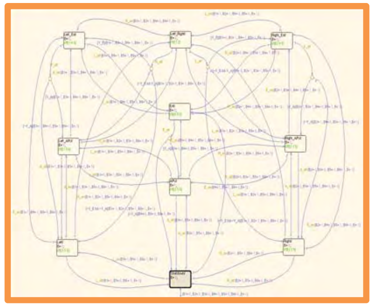
Physical system

⊗



Embedded system

⊗

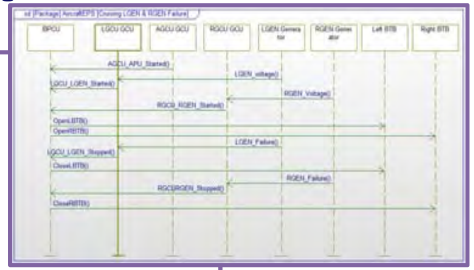


Controller

Preview of Our Strategy:
Contracts Provide Formal Support to CPS Design

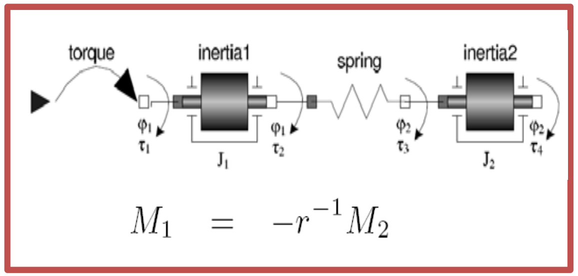
Structure and **formalize** requirements
- Component/Environment
- Functional/Safety/Timing

1.Reliability
2. Safety
3. Performance
4. Cost (e.g. energy, weight,…)



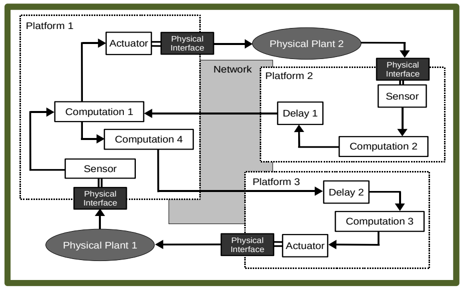
Requirements

Conjunction: Satisfy? ∧



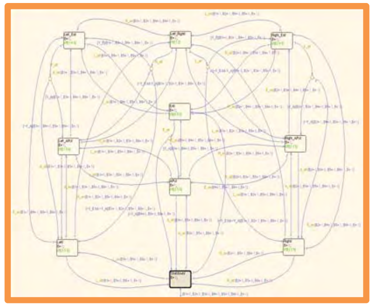
Physical system

⊗



Embedded system

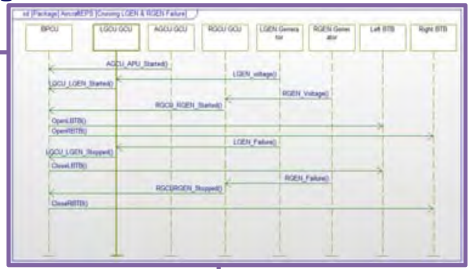
⊗



Controller

Preview of Our Strategy:
Contracts Provide Formal Support to CPS Design

1.Reliability
2. Safety
3. Performance
4. Cost (e.g. energy, weight,…)



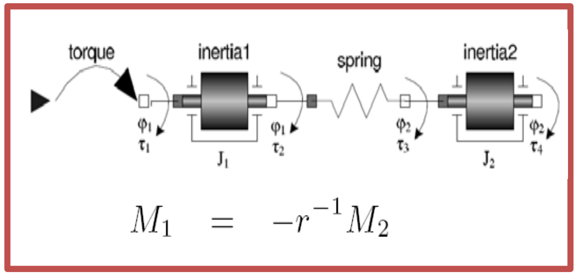
Requirements

Conjunction:
Satisfy?

$\wedge$

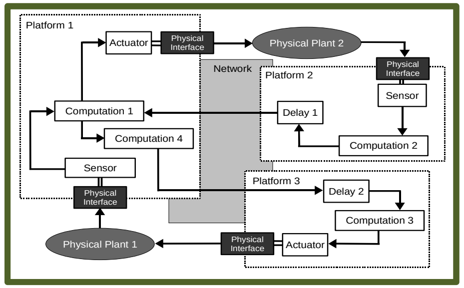
Refinement:
Satisfy? Replace?

$\preccurlyeq$



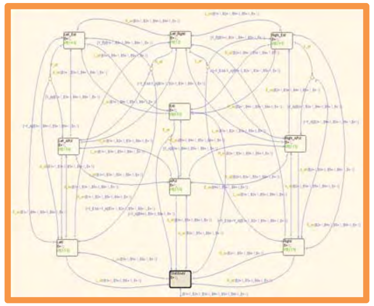
Physical system

$\otimes$



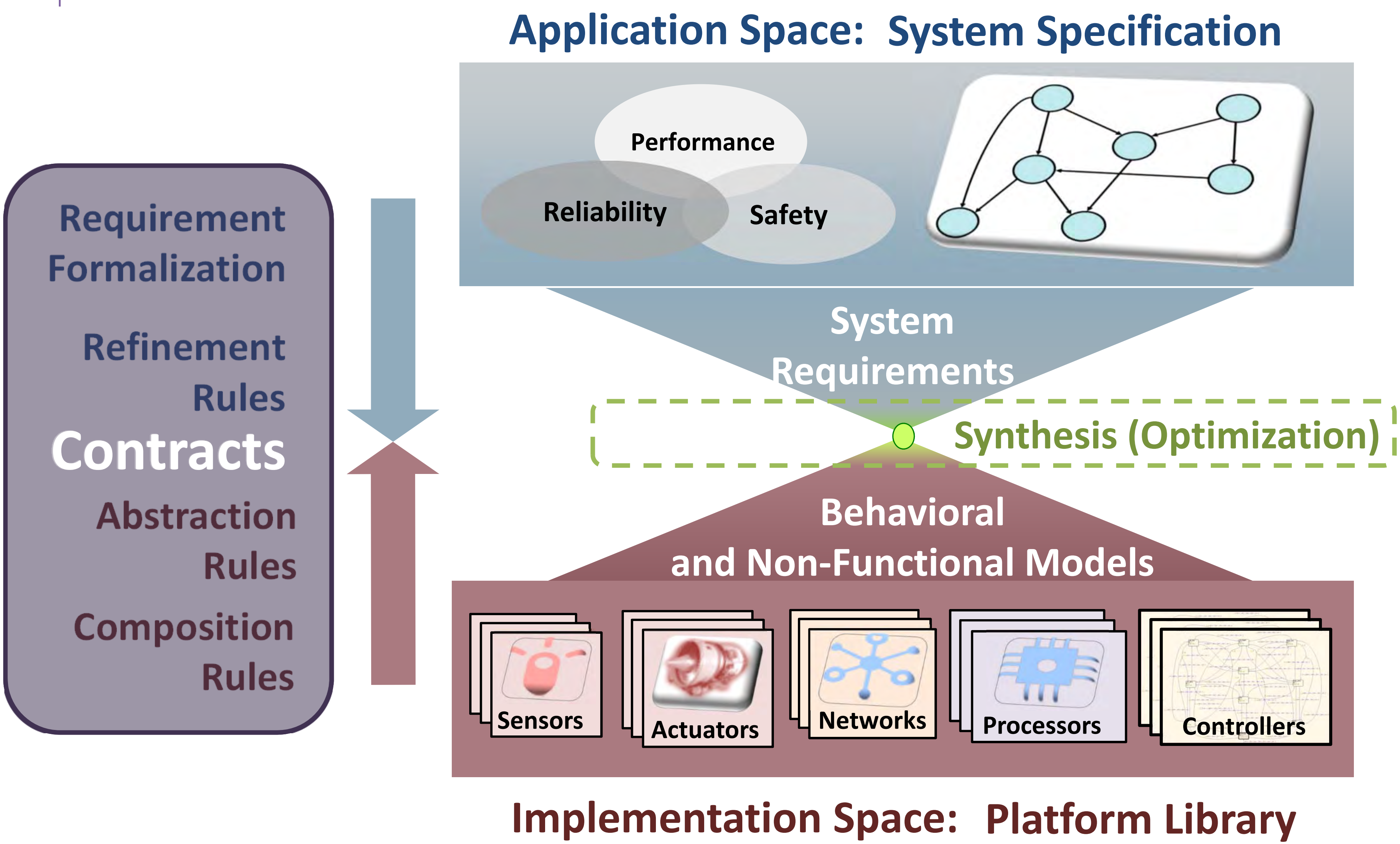
Embedded system

$\otimes$



Controller

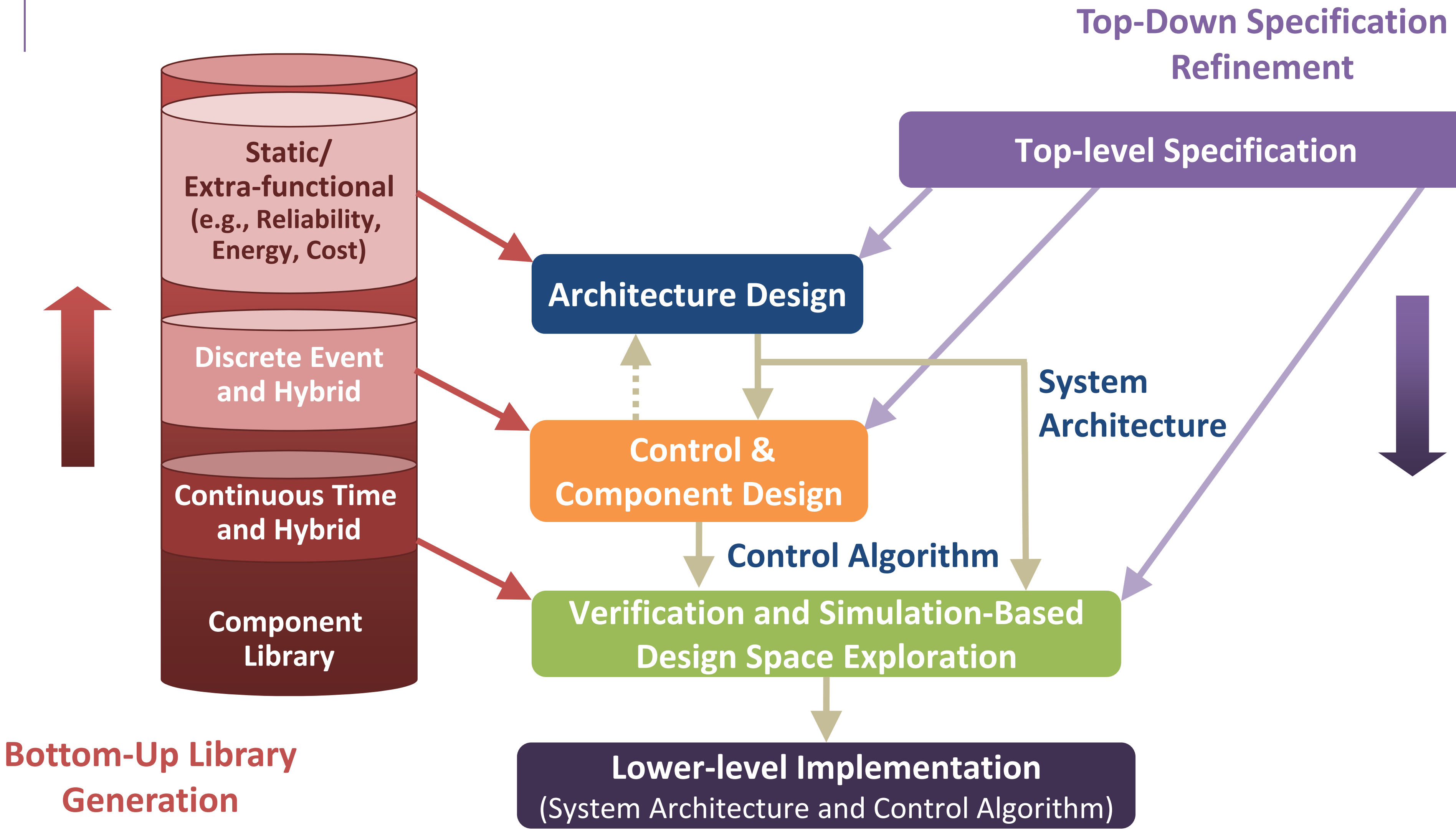
Preview of Our Strategy: Combine Platform-Based Design With Contracts



Outline

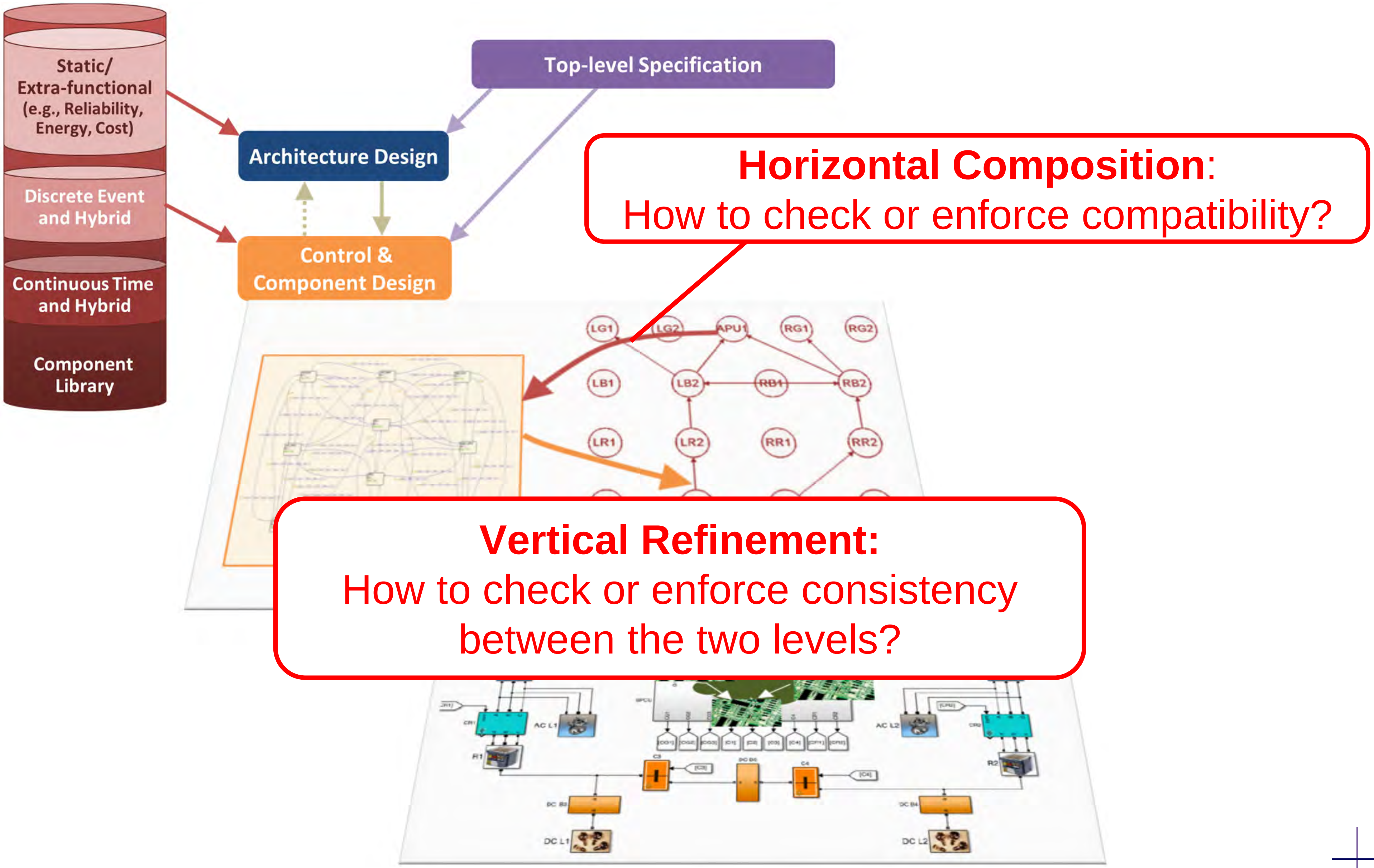
- Platform-based design methodology with contracts
- Requirement formalization
- Architecture design
- Control design
- Summary and future directions

The Structure of the Methodology: A Meet-in-the-Middle Approach



"Methodology and Tools for Next Generation Cyber-Physical Systems: The iCyPhy Approach," *INCOSE* 2015

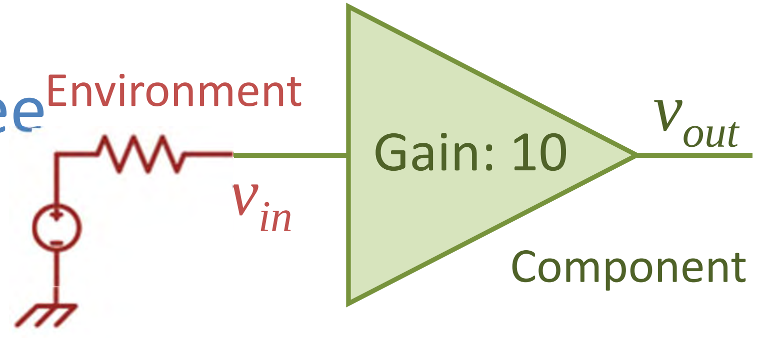
The Structure of the Methodology: Horizontal and Vertical Integration Steps



Formalizing the Methodology: Assume/Guarantee (A/G) Contracts

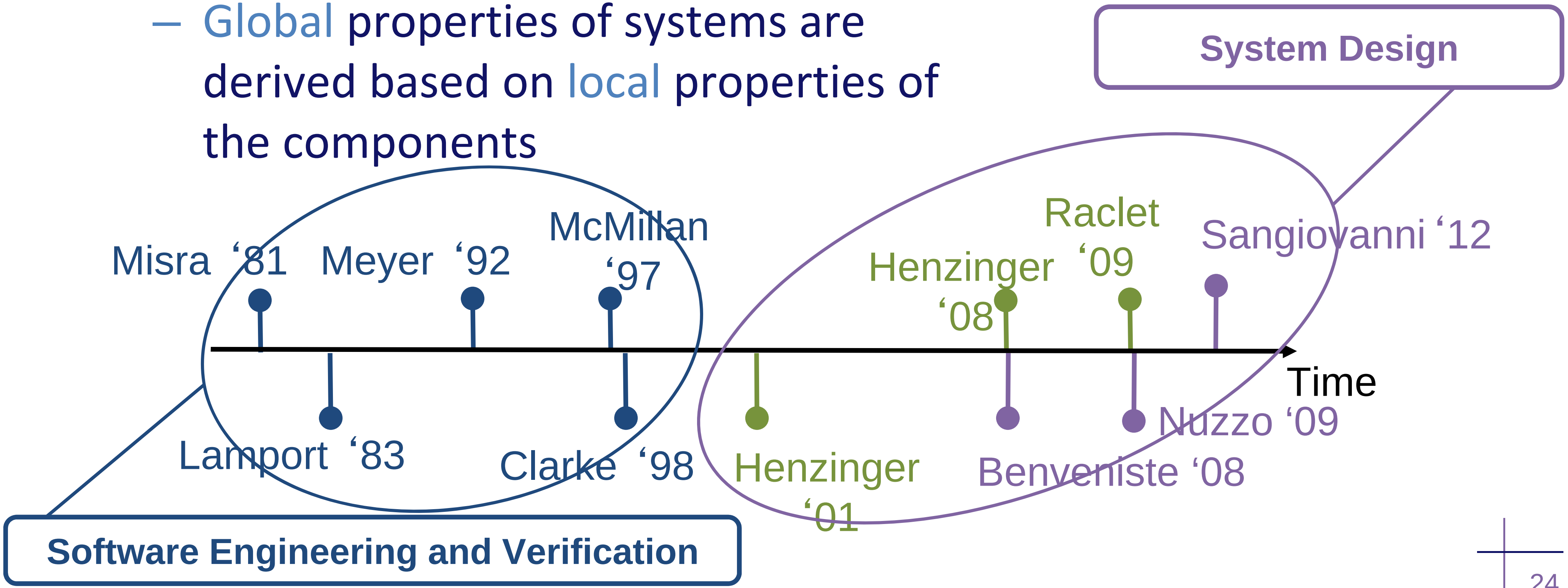
Contracts here are Assume-Guarantee pairs

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



Assume/Guarantee (A/G) Contracts: Mathematical Formulation

Set $V = I \cup O$ of **variables**
Set A of **assumptions**
Set G of **guarantees**

An **implementation** M **satisfies** a contract if $\boldsymbol{M \cap A \subseteq G}$

An **environment** E **satisfies** a contract if $\boldsymbol{E \subseteq A}$

Refinement $C_1 \preccurlyeq C_2$

$A_1 \supseteq A_2 \qquad G_1 \subseteq G_2$

Composition $C_1 \otimes C_2$

$A = (A_1 \cap A_2) \cup \neg G_1 \cup \neg G_2$
$G = G_1 \cap G_2$

(A, G) is **compatible** iff $A \neq \emptyset$
(A, G) is **consistent** iff $G \neq \emptyset$

Conjunction $C_1 \wedge C_2$

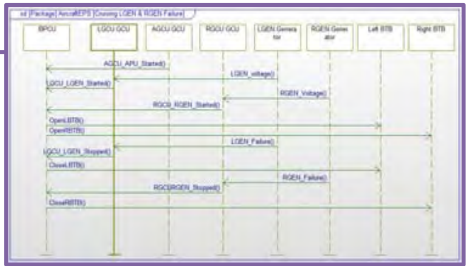
Independent Refinement

$$\left.\begin{array}{r}(C_1, C_2)\ \text{compatible} \\ C_i' \preceq C_i\end{array}\right\} \Rightarrow \left\{\begin{array}{l}(C_1', C_2')\ \text{compatible} \\ C_1' \otimes C_2' \preceq C_1 \otimes C_2\end{array}\right.$$

[Benveniste, Caillaud, Nickovic, Passerone, Raclet, Reinkemeier, Sangiovanni-Vincentelli, et al., 2012]

Contracts for Formalizing, Analyzing, and Propagating Requirements

1.Reliability
2. Safety
3. Performance
4. Cost (e.g. energy, weight,…)



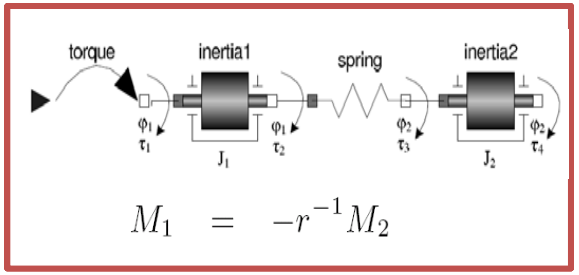
Requirements

Structure and **formalize**
- Component/Environment
- **Viewpoints:** Functional/Safety/Timing

$\wedge$ — **Conjunction**: Satisfy?

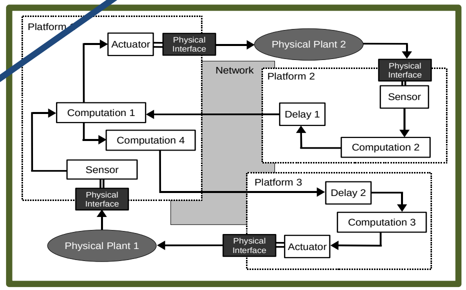
$\preccurlyeq$ — **Refinement**: Satisfy? Replace?

Composition: Compatible?



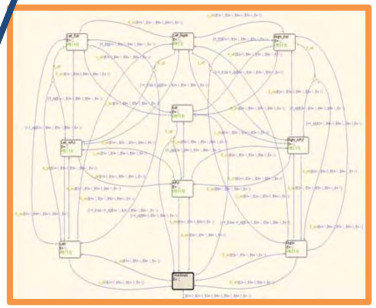
Physical system

$\otimes$



Embedded system

$\otimes$

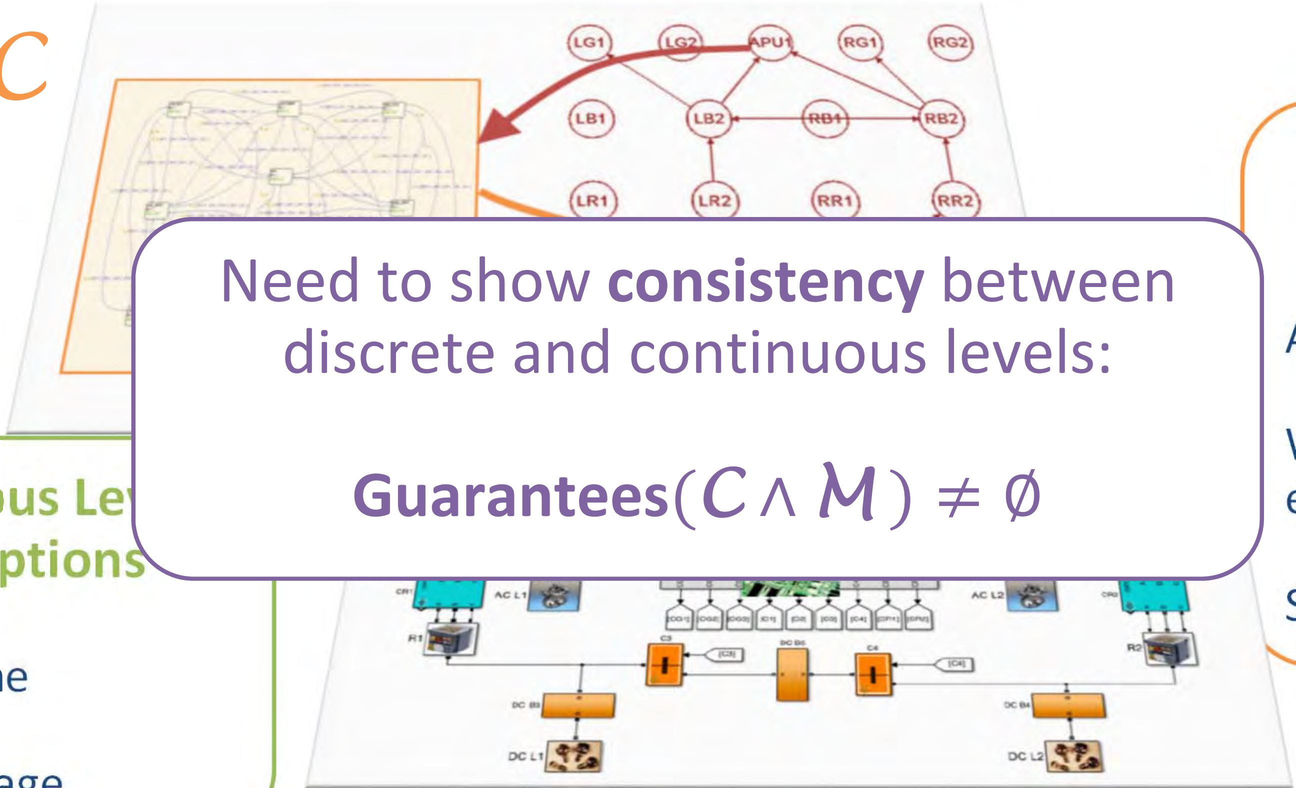


Controller

Horizontal and Vertical Contracts

- Horizontal contracts deal with components at the same level of abstraction
- Vertical contracts express assumptions and guarantees w.r.t. another level of abstraction

$\mathcal{C}$



Discrete Level Assumptions

Actuation delay

Worst case execution time

Sensor accuracy

$\mathcal{M}$

Need to show **consistency** between discrete and continuous levels:

$$\textbf{Guarantees}(\mathcal{C} \wedge \mathcal{M}) \neq \emptyset$$

Continuous Le[vel] Assumptions

Reaction time

Resource usage

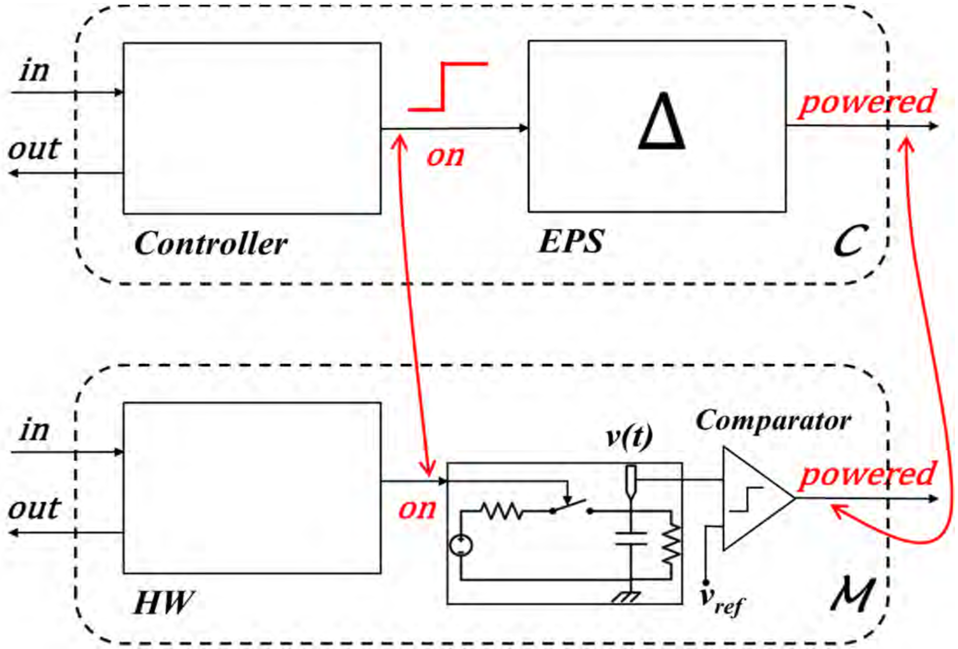
"Methodology for the Design of Analog Integrated Interfaces Using Contracts," ***IEEE Sensors J.*****, 2012**

Formalizing Vertical Contracts

- The specification contract $\mathcal{C}$ and implementation contract $\mathcal{M}$ are captured by heterogeneous architectural decompositions and behavior formalisms

- Satisfaction of all requirements and viewpoints depends on how system functionalities are mapped into execution platform and physical system

$$\mathcal{C} = \bigwedge_{k \in K} \left(\bigotimes_{i \in I_k} \mathcal{C}_{ik} \right)$$



$$\mathcal{M} = \bigotimes_{j \in J} \left(\bigwedge_{n \in N_j} \mathcal{M}_{jn} \right)$$

"A Platform-Based Methodology with Contracts and Related Tools for the Design of Cyber-Physical Systems," *Proceedings of the IEEE*, 2015

Directly checking $\mathcal{M} \preceq \mathcal{C}$ is not effective or compositional!

Formalizing Vertical Contracts

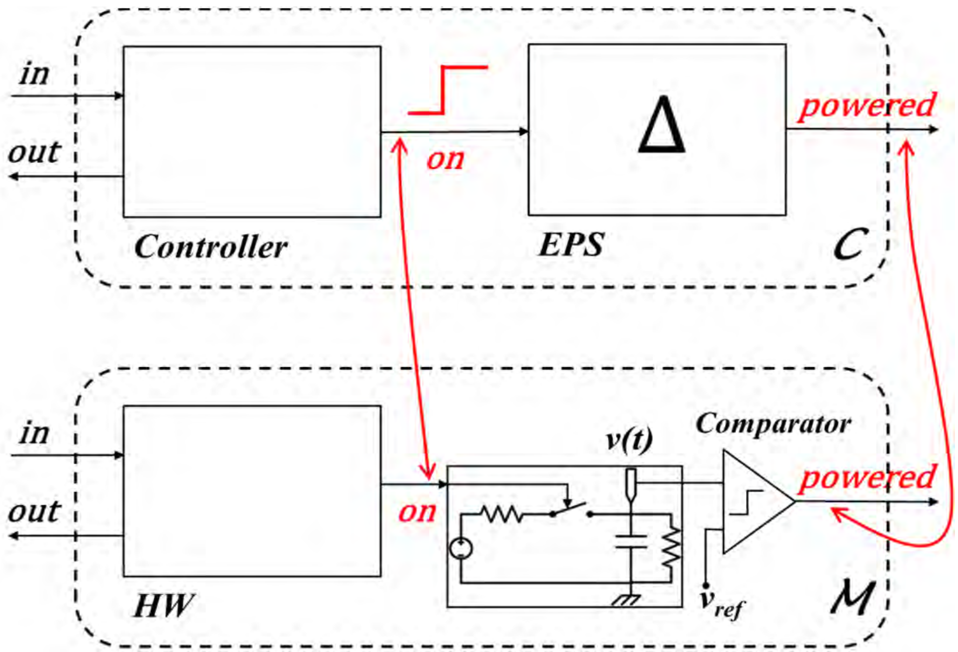
- Consider the vertical contract

$$\mathcal{C} \wedge \mathcal{M}$$

  - Refines $\mathcal{C}$ by construction
  - Must be consistent!
  - Discharges assumptions made by the specification layer using the guarantees of the implementation layer and vice versa

- A vertical contract specifies the conjunction of a model and its vertical refinement by connecting them through a mapping, e.g.,
  - by synchronizing pairs of events [METROPOLIS, Balarin et al., 2003]
  - by conjunction of constraints

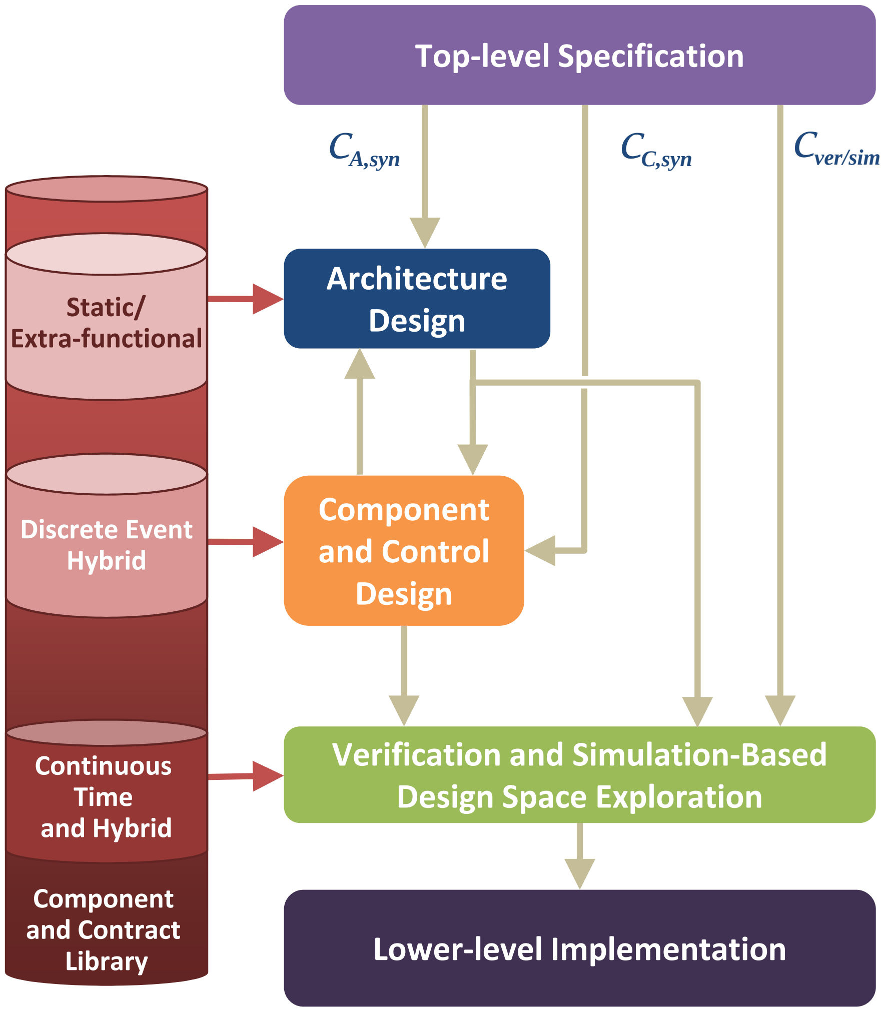
$$\mathcal{C} = \bigwedge_{k \in K} \left(\bigotimes_{i \in I_k} \mathcal{C}_{ik} \right)$$



$$\mathcal{M} = \bigotimes_{j \in J} \left(\bigwedge_{n \in N_j} \mathcal{M}_{jn} \right)$$

Vertical contracts support a richer set of refinements, e.g., synthesis and optimization-based methods

Methodology and Tools: Requirement Formalization



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.

CHASE: An Experimental Platform for Contract-Based Requirement Engineering

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.

Pattern-Based Contract Specification Language /

CHASE (Contract-based Heterogeneous Analysis and System Exploration)

Mixed Integer Linear Contracts (e.g., Steady-state, Topological)

$$\sum_{i=1}^{n_{rec}} M^{rd}_{i,j} \geq \sum_{i=1}^{n_{load}} M^{dl}_{j,i}, \quad \sum_{i=1}^{n_{rec}} M^{rd}_{i,j} \geq \sum_{i=1}^{n_{dcb}} M^{dd}_{j,i}$$

Linear Temporal Logic [Pnueli'77] **Contracts (e.g., Safety)**

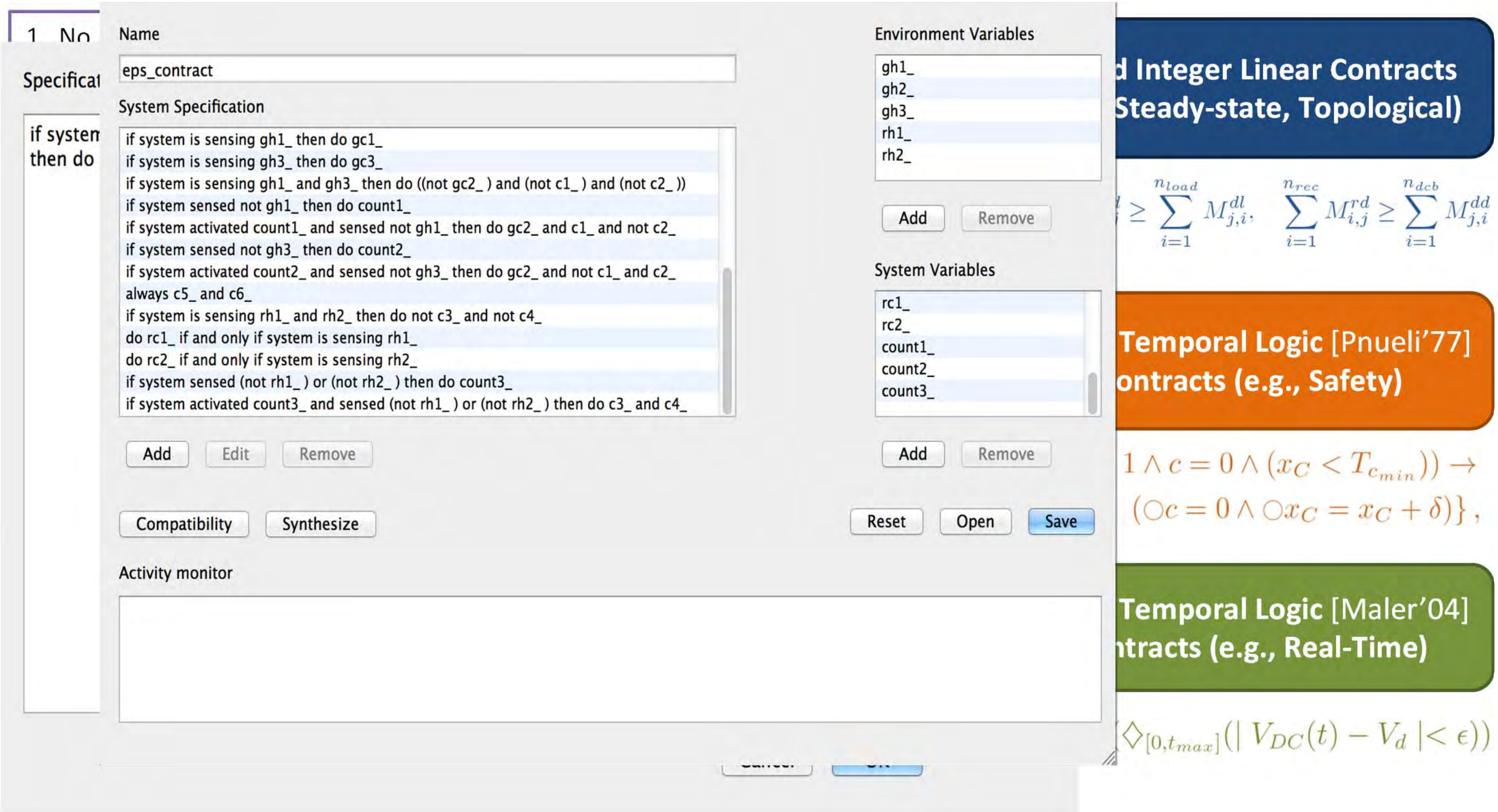
$$\Box\{(\tilde{c} = 1 \wedge c = 0 \wedge (x_C < T_{c_{min}})) \rightarrow (\bigcirc c = 0 \wedge \bigcirc x_C = x_C + \delta)\},$$

Signal Temporal Logic [Maler'04] **Contracts (e.g., Real-Time)**

$$\Box_{[T_i,\infty)}(\Diamond_{[0,t_{max}]}(| V_{DC}(t) - V_d | < \epsilon))$$

State-of-the-art tools for Requirement Management (IBM DOORS) lack formal semantics …

CHASE: An Experimental Platform for Contract-Based Requirement Engineering



...

Integrating with state-of-the-art tools for Requirement Management (DOORS) and Natural Language Processing (WATSON)

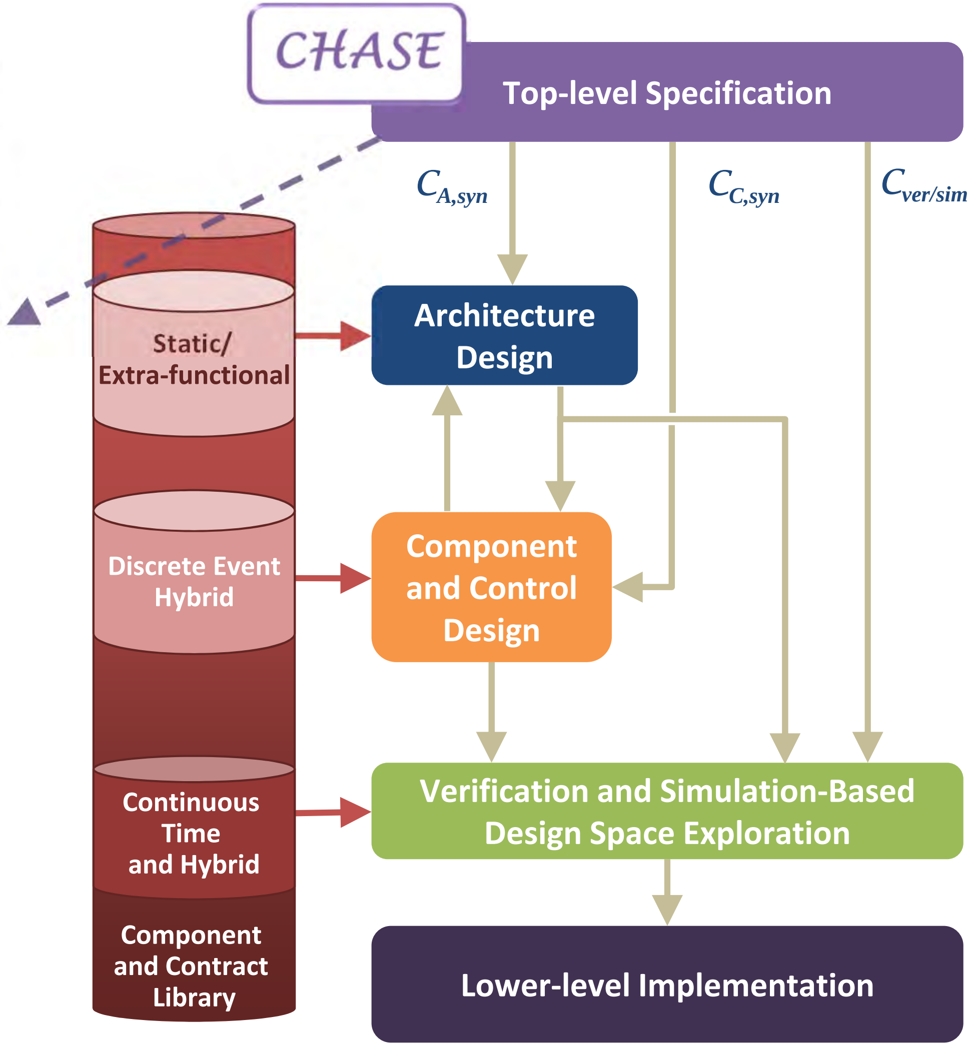
Methodology and Tools: Architecture Design

$$\Box\{(\tilde{c} = 1 \wedge c = 0 \wedge (x_C < T_{c_{min}})) \rightarrow (\bigcirc c = 0 \wedge \bigcirc x_C = x_C + \delta)\},$$

$$\Box\{(\tilde{c} = 1 \wedge c = 0 \wedge (x_C \geq T_{c_{min}})) \rightarrow (\bigcirc c = 1 \vee \bigcirc x_C = x_C + \delta)\},$$

$$\sum_{i=1}^{n_{rec}} M_{i,j}^{rd} \geq \sum_{i=1}^{n_{load}} M_{j,i}^{dl}, \quad \sum_{i=1}^{n_{rec}} M_{i,j}^{rd} \geq \sum_{i=1}^{n_{dcb}} M_{j,i}^{dd}$$

$$\Box_{[\tau_i,\infty)}(\Diamond_{[0,t_{max}]}(| V_{DC}(t) - V_d | < \epsilon))$$



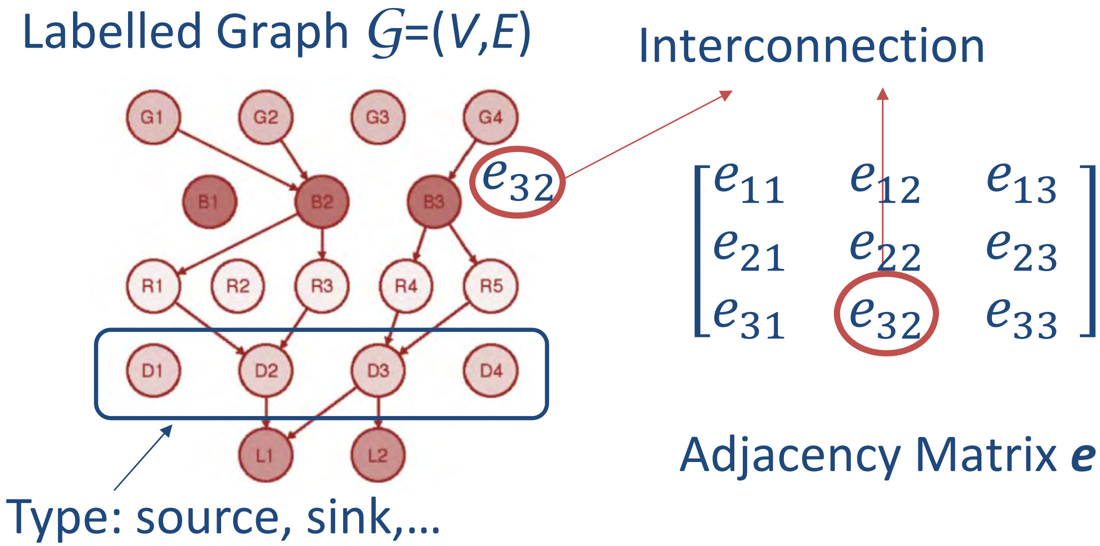
1. No AC bus shall be simultaneously powered by more than one AC source.
2. The aircraft electric power system shall provide power with the following characteristics: 115 +/- 5 V (amplitude) and 400 Hz (frequency) for AC loads and 28 +/-2 V for DC loads.
3. The failure probability at an essential load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

Architecture Exploration Problem: Component Library and Attributes

- **Variables:** u, x, y – input, internal, output
- **Parameters:** $\kappa = (s, p)$ – discrete, continuous
- **Behavioral Model:** $\mathcal{F}(u, x, y, \kappa) = 0$ – e.g., differential algebraic equations (DAE)
- **Extra-Functional Model:** compact maps providing energy, performance, cost, reliability...
- **Terminals:** Logical (input/output)
  Physical (hydraulic, thermal, electrical,...)
- **Contracts:** Sets of behaviors on variables, parameters, and terminals

Type
(function in the system)

Architecture Exploration: Problem Statement



- An assignment over the edge variables defines a topology
- Nodes and edges of a topology are labelled with the attributes of the components implementing them

Given a library $\mathcal{L}$, select **topology** $\mathcal{T}$ **(number, type, and interconnections)** and **component dimensions** to implement each node and edge in $\mathcal{T}$ while satisfying a set of **contracts** and minimizing a **cost**

Architecture Exploration: Mathematical Formulation

System-level **specification contract** C

Determine $\boldsymbol{\kappa}^*$ such that $C \wedge \mathcal{M}$ is **consistent**, i.e., there exists an implementation satisfying both C and $\mathcal{M}$

Implementation contract $\mathcal{M}$

$$\min_{p \in \mathcal{P}, s \in \mathcal{S}} \quad C(s,p)$$

s. t.

functional

$$r_k(s, p, x_\infty, y_\infty) \leq 0$$

$$r_m(s, p, x_\infty, y_\infty) \leq 0$$

extra-functional

$$\mathcal{F}(u, x, y, s, p, x_0) = 0$$

$$\forall u(t) \in \mathcal{U}, \forall x_0 \in \mathcal{X}_0$$

architecture

$$\bigwedge_{i \in V \cup E, j \in \mathcal{L}_i} {}_E \; s_{ij} \mathcal{F}_j(u_j, x_j, y_j, d_j, p_j) \qquad \sum_{j \in \mathcal{L}_i} s_{ij} = 1, \quad \forall i \in V \cup E$$

Architecture Exploration: Mathematical Formulation

System-level **specification contract** C

Determine $\boldsymbol{\kappa}^*$ such that $C \wedge \mathcal{M}$ is **consistent**, i.e., there exists an implementation satisfying both C and $\mathcal{M}$

$$\min_{p \in \mathcal{P}, s \in \mathcal{S}} \quad C(s,p)$$

s. t.

functional

$$r_k(s,p,x_\infty,y_\infty) \leq 0$$

$$r_m(s,p,x_\infty,y_\infty) \leq 0$$

extra-functional

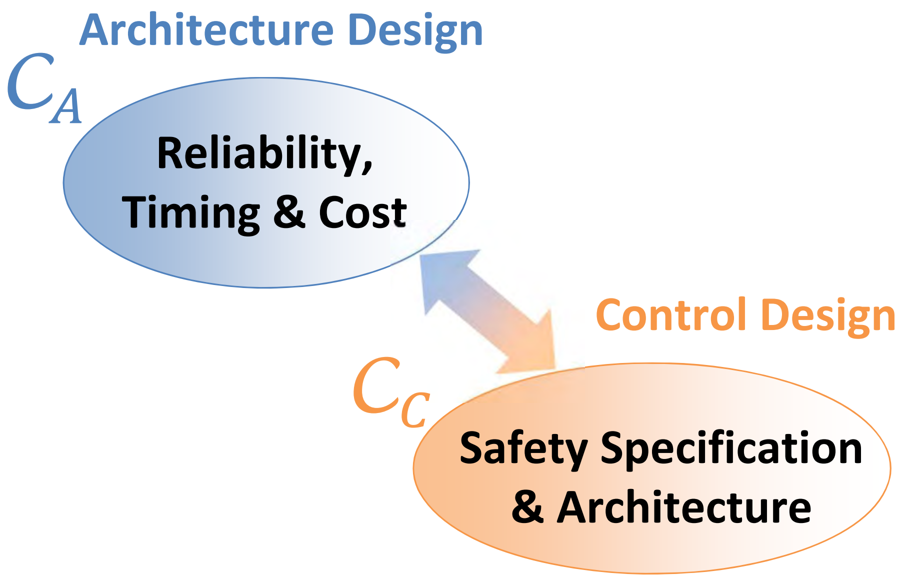
$$\mathcal{F}(u,x,y,s,p,x_0) = 0$$

$$\forall u(t) \in \mathcal{U} \ \forall x_0 \in \mathcal{X}_0$$

*Often an **intractable problem***

- ***Large** number of **discrete** alternatives*
- ***Expensive** analyses or **simulations** to accurately estimate performance and cost*
- ***Complex** non-linear models, often not available in closed analytic form*

EPS Example:
Deriving the Architecture Requirements



$$C_A \otimes C_C \preccurlyeq C_S$$

Failure probability:
$r_A \leq r_S$

Failure probability: r_S

Worst case "actuation delay":
$T^* \leq t_{max}$ **(bus unpowered time)**

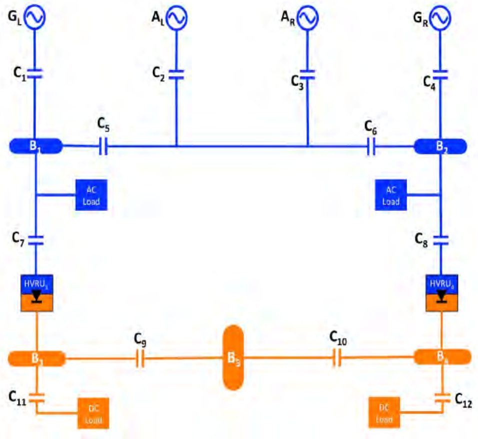
Proposition

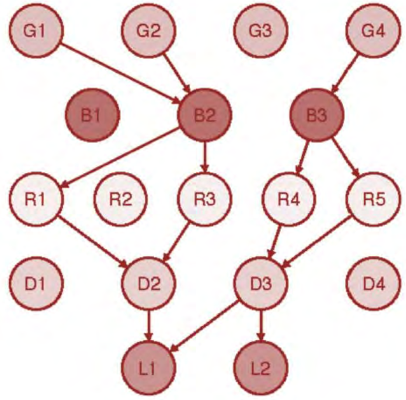
Given a ***system*** *contract C_S with* ***reliability requirement r_S****, if the* ***topology*** *implements C_A with a* ***reliability level $r_A \leq r_S$****, then there exists a* ***time T^**** *such that*

1) a ***centralized controller*** *implementing C_C with a* ***reliability level r_S*** *and* ***maximum bus unpowered time $t_{max} \geq T^*$*** *is realizable*

2) the controlled system satisfies the ***system contract*** *C_S*

EPS Example: Optimized Selection of Reliable and Cost-Effective Architectures





Generating **symbolic probability constraints** on a **parametrized graph** is not efficient!

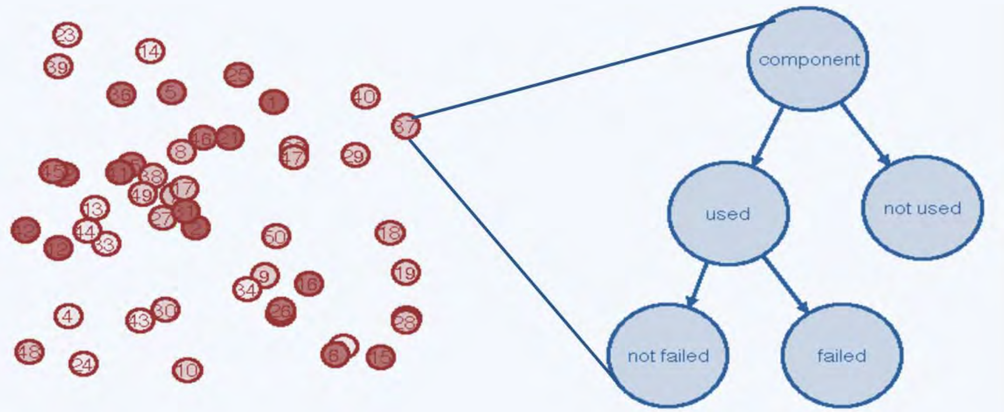
Objective: minimize node and edge cost

$$\sum_{i=1}^{|V|} \delta_i c_i + \sum_{i=1}^{|V|} \sum_{j=i+1}^{|V|} (e_{ij} \vee e_{j1})\tilde{c_{ij}}$$

Interconnection: there exists at least (most) one connection from a node in *L* and a node in *D*

$$\sum_{i=1}^{|D|} e_{l_j d_i} \geq (\leq)\ 1 \quad \forall\, j \in \mathbb{N}:\ 1 \leq j \leq |L|$$

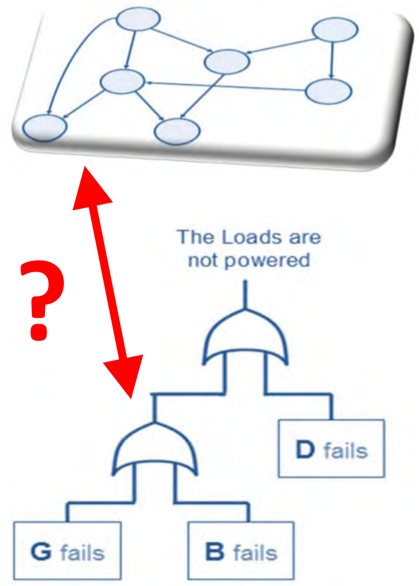
Reliability:
the probability that a sink gets disconnected from a source should be less than *r**



Optimized Selection of Reliable and Cost-Effective Architectures

Reliability as a function of the interconnection structure and component failure probabilities

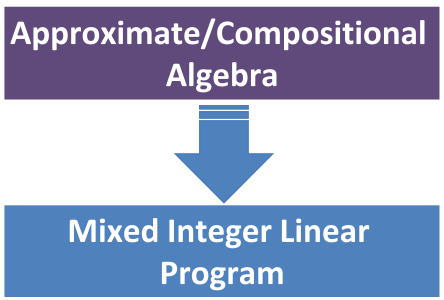
- Expensive: Exact analysis is NP-hard [Lucet '97]
- Non-compositional: computed via Fault-Tree Analysis (FTA) or Reliability Block Diagrams (RBD), based on modules that are not directly linked to system components [Kaiser '03]



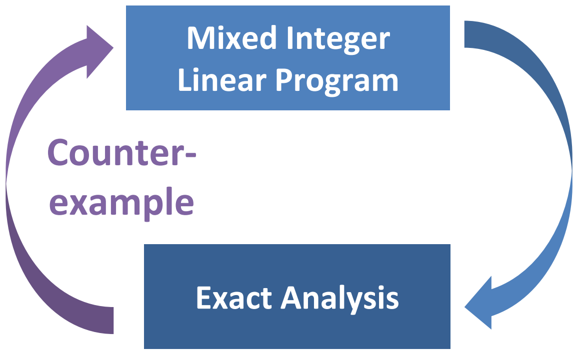
Monolithic Optimization with Approximate Analysis

Mixed Integer Linear Programming With Approximate Reliability

Iterative Optimization with Exact Analysis



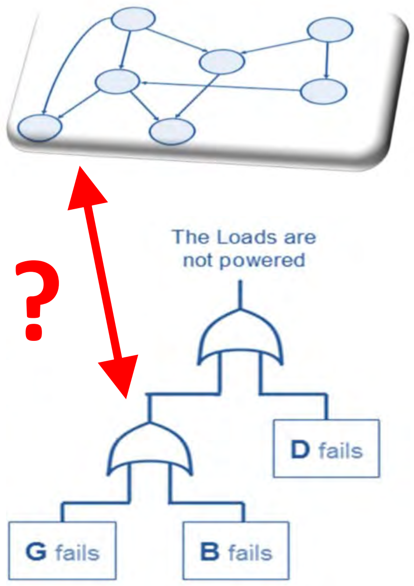
Mixed Integer Linear Programming Modulo Reliability



Optimized Selection of Reliable and Cost-Effective Architectures

Reliability as a function of the interconnection structure and component failure probabilities

- Expensive: Exact analysis is NP-hard [Lucet '97]
- Non-compositional: computed via Fault-Tree Analysis (FTA) or Reliability Block Diagrams (RBD), based on modules that are not directly linked to system components [Kaiser '03]



Monolithic Optimization with Approximate Analysis

Mixed Integer Linear Programming With Approximate Reliability

Approximate/Compositional Algebra

↓

Mixed Integer Linear Program

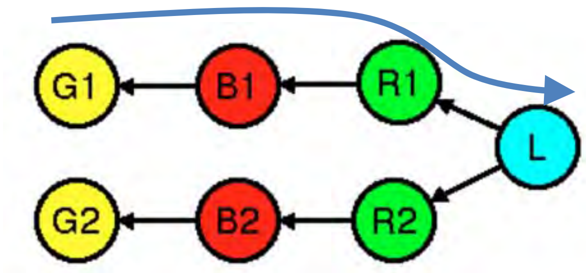
Approximation introduces a number of **linear** probability constraints and auxiliary variables **polynomial** in the number of nodes V and types m ($O(V^3 m)$)

Soundness: If the algorithm finds a solution, does it actually satisfy the requirements?
Completeness: If there is a solution, can the algorithm actually find it?

Mixed Integer Linear Programming With Approximate Reliability (MILP-AR)

Components contribute to system reliability based on their degree of redundancy h and failure probability p

- **Functional link F_i**: set of paths from any source node to a sink v_i; let c_{ij} be the number of components of *type j* used in at least one path of F_i
- **Degree of redundancy h_{ij}: $h_{ij} = c_{ij}$** if type j is ***maximally interconnected***
  $\min\{c_{ij} \mid c_{ij} > 1\}$ otherwise



2 2 2 1 Degree of Redundancy

Exact: $p + 9p^2 + O(p^3)$

Approximate: $p + 6p^2$

$$\tilde{r}_i = \sum_{j \in I_i} c_{ij} p_j^{h_{ij}}$$

h_{ij}: degree of redundancy for type j in F_i

c_{ij}: number of components of type j in F_i

Theorem 1. There exists a theoretical bound to the approximation error

- **m: number of types involved in F_i**
- **h: minimum of $\{h_{ij} \mid h_{ij} > 1\}$ over all redundant types j in $\{1,\ldots,m\}$**

$$\frac{\tilde{r}}{r} \geq \frac{h}{m^{h-1}}$$

MILP-AR: Lower Bound on Approximate Reliability Measure

$$\frac{\tilde{r}}{r} \geq \frac{h}{m^{h-1}}$$

- Approximation is "conservative" for maximally interconnected types
  - No redundancy: If h_i=1 for at least one type i in F, then $\tilde{r}/r \geq 1$
  - Maximum redundancy

$$\tilde{r}^{max} = \sum_{i=1}^{m} c_i p_i^{c_i} \geq \sum_{i=1}^{m} p_i^{c_i} \geq r^{max}$$

- Minimum $\tilde{r}/r$ is achieved when F is a tree with h independent paths from sources to the sink
  - All types in F have the same (minimum) redundancy: $h_i = h$ for all i
  - Paths in F are independent: no node must be shared other than the sink

$$\tilde{r} = \sum_{i=1}^{m} h p_i^h \qquad r = (1 - (1 - \bar{p})^m)^h$$

$\boldsymbol{h_i}$: degree of redundancy for type i in F

$\boldsymbol{c_i}$: number of components of type i in F

$\boldsymbol{p_i}$: failure probability for components of type i in F

$$\bar{p} = 1 - \left(\prod_{i=1}^{m}(1 - p_i)\right)^{\frac{1}{m}}$$

MILP-AR: Lower Bound on Approximate Reliability Measure

$$\frac{\tilde{r}}{r} \geq \frac{h}{m^{h-1}}$$

$$\tilde{r} = \sum_{i=1}^{m} h p_i^h \qquad r = (1-(1-\bar{p})^m)^h$$

h_i: degree of redundancy for type *i* in *F*

c_i: number of components of type *i* in *F*

$$\bar{p} = 1 - \left(\prod_{i=1}^{m}(1-p_i)\right)^{\frac{1}{m}}$$

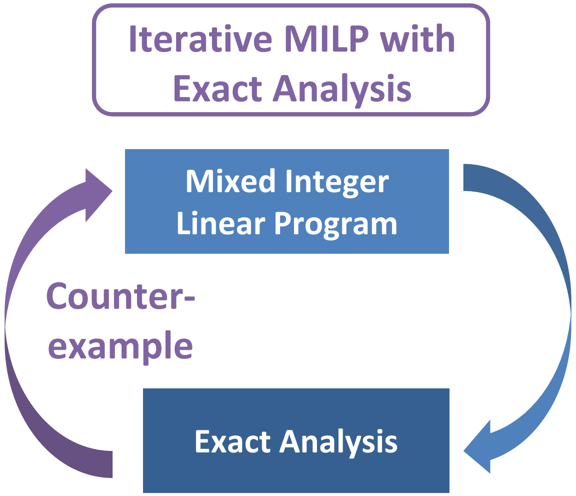
- Minimum $\tilde{r}/r$ is achieved when *F* is a tree with *h* independent paths from sources to the sink
  - There exists p^* independent of *h* or *m* such that, $\forall\, p_i \leq p^*$:

$$\tilde{r} = \sum_{i=1}^{m} h p_i^h \geq m h \bar{p}^h$$

  - … and $\tilde{r}$ achieves its minimum for $p_1 = \cdots = p_m = \bar{p}$
- Then, we conclude $\frac{\tilde{r}}{r} = \frac{m h \bar{p}^h}{(1-(1-\bar{p})^m)^h} \geq \frac{m h \bar{p}^h}{\bar{p}^h m^h} = \frac{h}{m^{h-1}}$

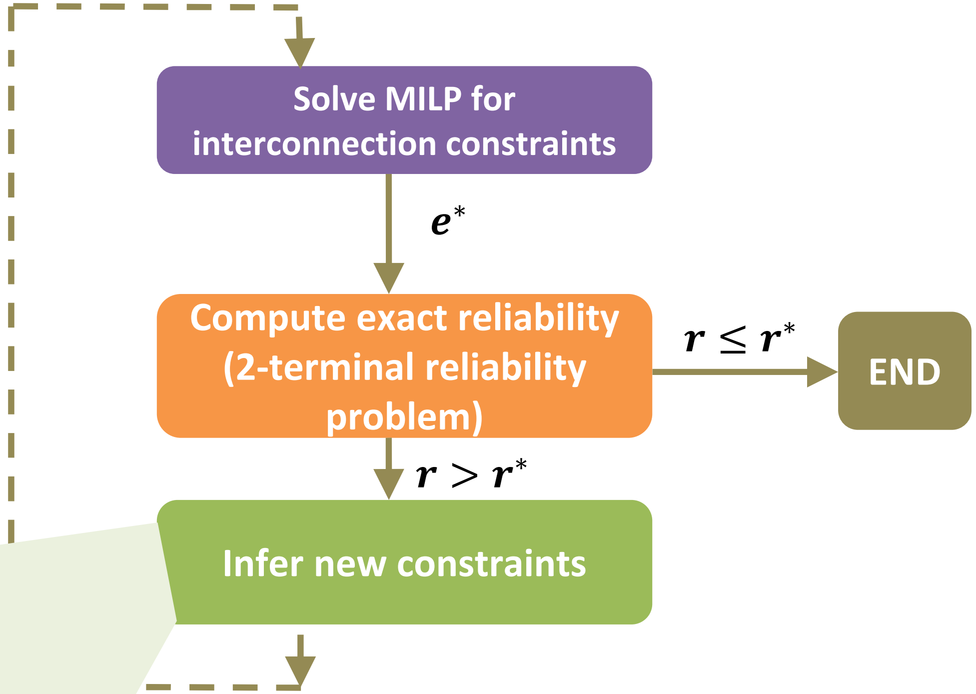
Theorem 2. *For a given library MILP-AR is **sound** and **complete** within the bounds of the approximate reliability measure*

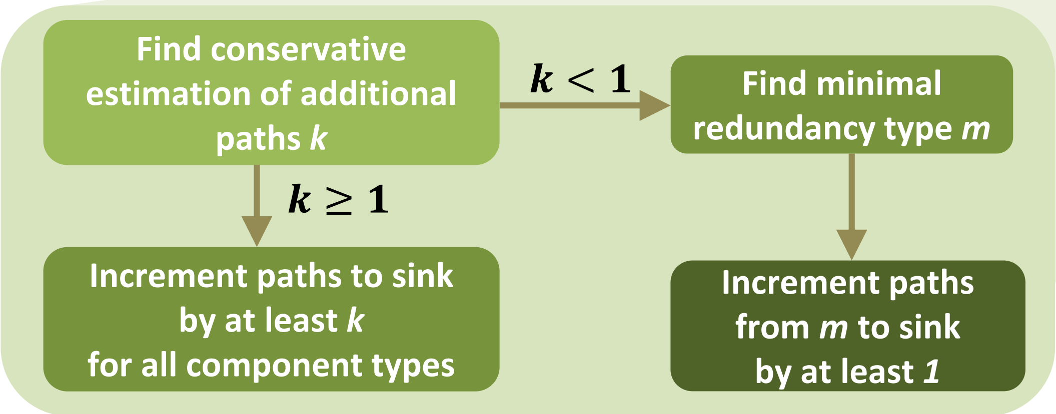
Mixed Integer Linear Programming Modulo Reliability (MILP-MR)



Avoid symbolic probability analysis on a parametrized graph

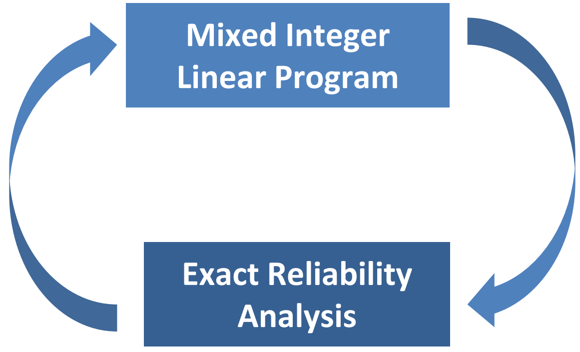
Perform exact numeric analysis on fixed graph:
- only when needed
- on smaller graph instances





Goal: Decrease the number of iterations

MILP-MR: Soundness and Completeness

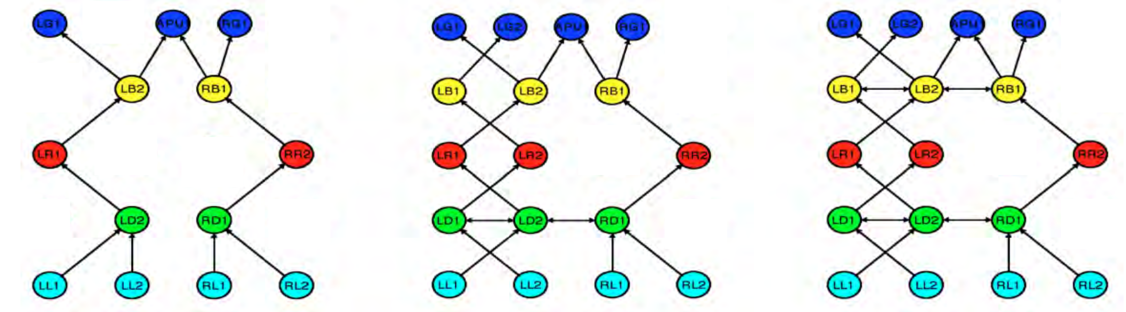


Theorem 3. *Given a library $\mathcal{L}$, if the MILP solver is **sound** and **complete** on its problem instances, then MILP-MR is sound and complete*

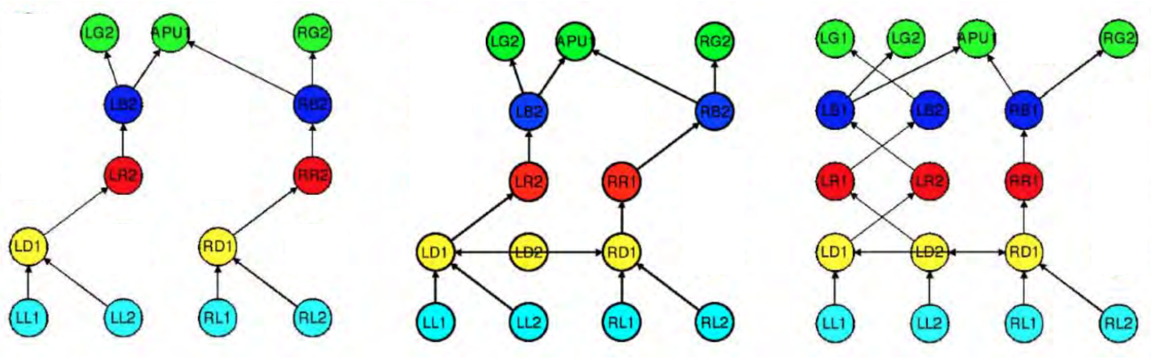
- MILP-MR terminates since the number of components is finite
  - The sequence of costs c_k is non-decreasing: nodes and edges may only increase at each iteration
  - The sequence of failure probabilities r_k is non-increasing
- Sound: If MILP-MR returns an architecture, then it satisfies all the requirements: failure probability is checked by exact analysis
- Complete: if MILP-MR terminates with INFEASIBLE, then
  - either redundant paths are inconsistent with interconnection constraints…
  - …or all available redundant components/edges in the given library are exhausted

ARCHEX Allows Effective EPS Design Space Exploration

Generators	g (kW)	Loads	l (kW)	Components	c
LG1	70	LL1	30	Generator	g/10
LG2	50	LL2	10	Bus	2000
RG1	80	RL1	10	Rectifier	2000
RG2	30	RL2	20	Contactor	1000
APU	100				



1: $r = 6 \cdot 10^{-4}$ 2: $r = 2.8 \cdot 10^{-10}$ 3: $r = 0.8 \cdot 10^{-10}$



1 2 3

Generators, buses and rectifiers fail with probability 2×10^{-4}

MILP-MR run: $r^*=2 \times 10^{-10}$ 38 s on an Intel Core i7 2.8-GHz, 8-GB RAM

MILP-AR run: 38 s (70% is constraint generation)

	1	2	3
Required	2×10^{-3}	2×10^{-6}	2×10^{-10}
Approximate	6×10^{-4}	2.4×10^{-7}	7.2×10^{-11}
Exact	6×10^{-4}	3.5×10^{-7}	2.8×10^{-10}

Error within the predicted bound

MILP-MR With Redundant Path Inference Outperforms MILP-AR for Large Architectures

MILP-AR

$\|V\|$ (# Generators)	# Constraints	Setup time (s)	Solver time (s)
20 (4)	5290	27	11
30 (6)	24514	402	77
40 (8)	74258	3341	494
50 (10)	176794	18902	5059

MILP-MR infers the number of redundant paths

$\|V\|$ (# Generators)	#Iterations	Analysis time (s)	Solver time (s)
20 (4)	3	34	4.3
30 (6)	3	78	9
40 (8)	3	106	14
50 (10)	3	181	18
20 (4)	4	72	13
30 (6)	7	852	28
40 (8)	10	9118	58
50 (10)	14	39563	114

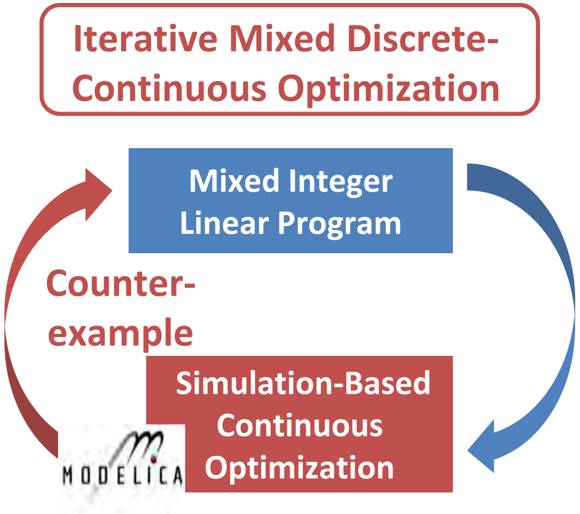
MILP-MR adds one path per iteration

MILP-AR: problems with several thousands of constraints, several hundred thousands variables, and a realistic number of generators (<10) can still be formulated and solved in a few hours

MILP-MR: Dramatic reduction in reliability analysis time by inferring redundant paths (3 min versus more than 1 day for 50 nodes)

"Optimized Selection of Reliable and Cost-Effective Cyber-Physical System Architectures," ***DATE'15***

Hybrid Optimization Scheme Explores up to 1.5-Million Configurations in < 2 hours



Library		Runtime Performance			
Size	Discrete Choices	Cost	Discrete Iterations	Simulations	Run Time (h)
Full Enumeration					
6	15,552	112.39	17	62,496	3.72
9	118,098	112.03	72	257,141	15.42
12	497,664	18-h Timeout			
15	1,518,750	21-h Timeout			
LearnCons					
6	15,552	112.39	2	5,812	0.36
9	118,098	112.03	4	13,329	0.83
12	497,664	111.63	6	21,418	1.34
15	1,518,750	111.14	9	31,874	1.91

- Sound and complete hybrid optimization scheme
  - Uses simulation-based version of Nelder-Mead algorithm to size the components of a given topology
  - Leverage conservation laws to generate new constraints that prune out the search space when a continuous solution is infeasible

- Applied to architecture exploration of an aircraft environment control system: > 1 order of magnitude reduction in run time with respect to full enumeration (Intel Xeon 3.59-GHz with 24-GB RAM)

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

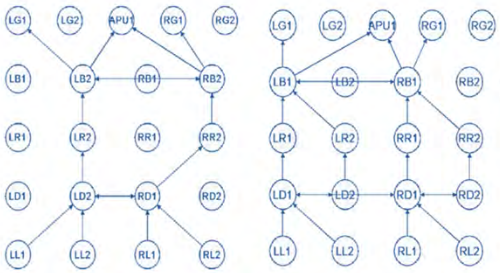
Methodology and Tools: Control Design

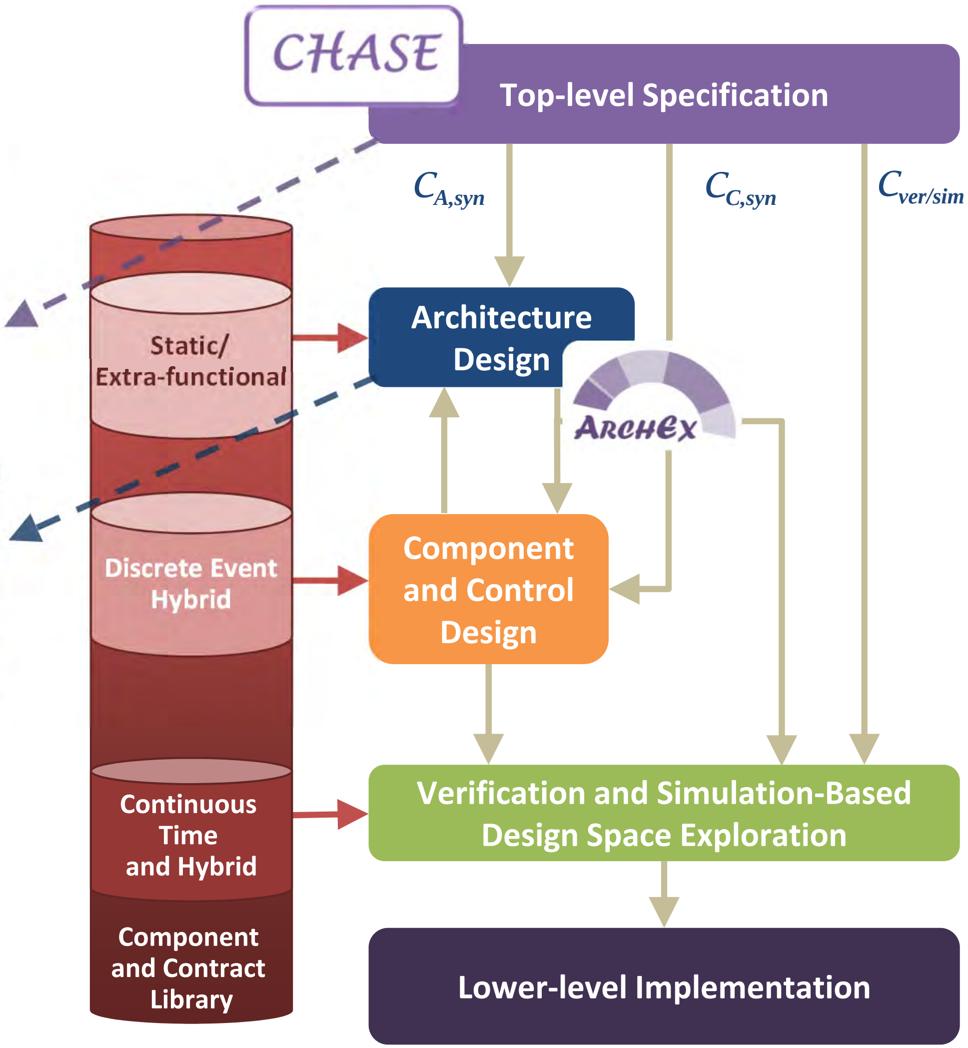
$$\square\{(\tilde{c}=1 \wedge c=0 \wedge (x_C < T_{c_{min}})) \rightarrow (\bigcirc c = 0 \wedge \bigcirc x_C = x_C + \delta)\},$$

$$\square\{(\tilde{c}=1 \wedge c=0 \wedge (x_C \geq T_{c_{min}})) \rightarrow (\bigcirc c = 1 \vee \bigcirc x_C = x_C + \delta)\},$$

$$\sum_{i=1}^{n_{rec}} M_{i,j}^{rd} \geq \sum_{i=1}^{n_{load}} M_{j,i}^{dl}, \quad \sum_{i=1}^{n_{rec}} M_{i,j}^{rd} \geq \sum_{i=1}^{n_{dcb}} M_{j,i}^{dd}$$

$$\square_{[T_i,\infty)}(\Diamond_{[0,t_{max}]}(|V_{DC}(t) - V_d| < \epsilon))$$





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: Control Design

G_C

Power Flow

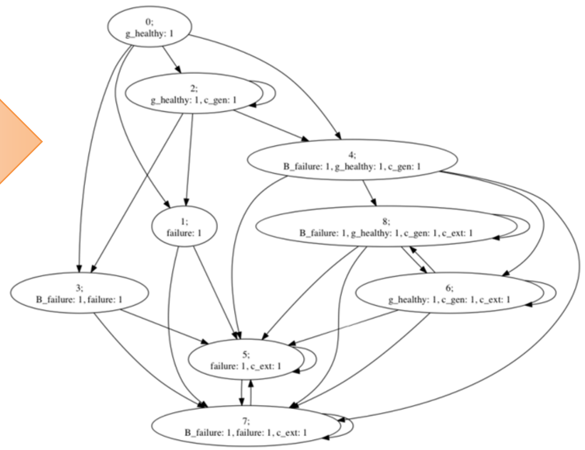
$$\square\left\{\bigwedge_{i\in\mathcal{I}}(e_i=1)\rightarrow\bigwedge_{i\in\mathcal{G}}\Big(P_{e_i}\geq\sum_{B\in\mathcal{B}} l_{i,B}\Big)\right\}$$

No Paralleling AC sources

$$\square\bigwedge_{i,j\in\mathcal{G}}\left\{\neg\bigwedge_{C\in p_{i,j}}(c=1)\right\}$$

Safety Specification & Architecture

Control Design



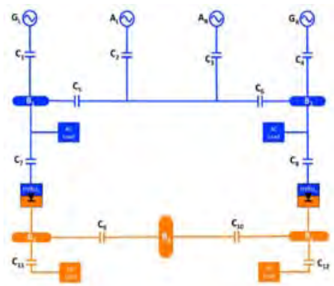
A_C

LTL Behavioral Models (e.g., contactor delays)

Environment Assumptions: Plant Reliability Level

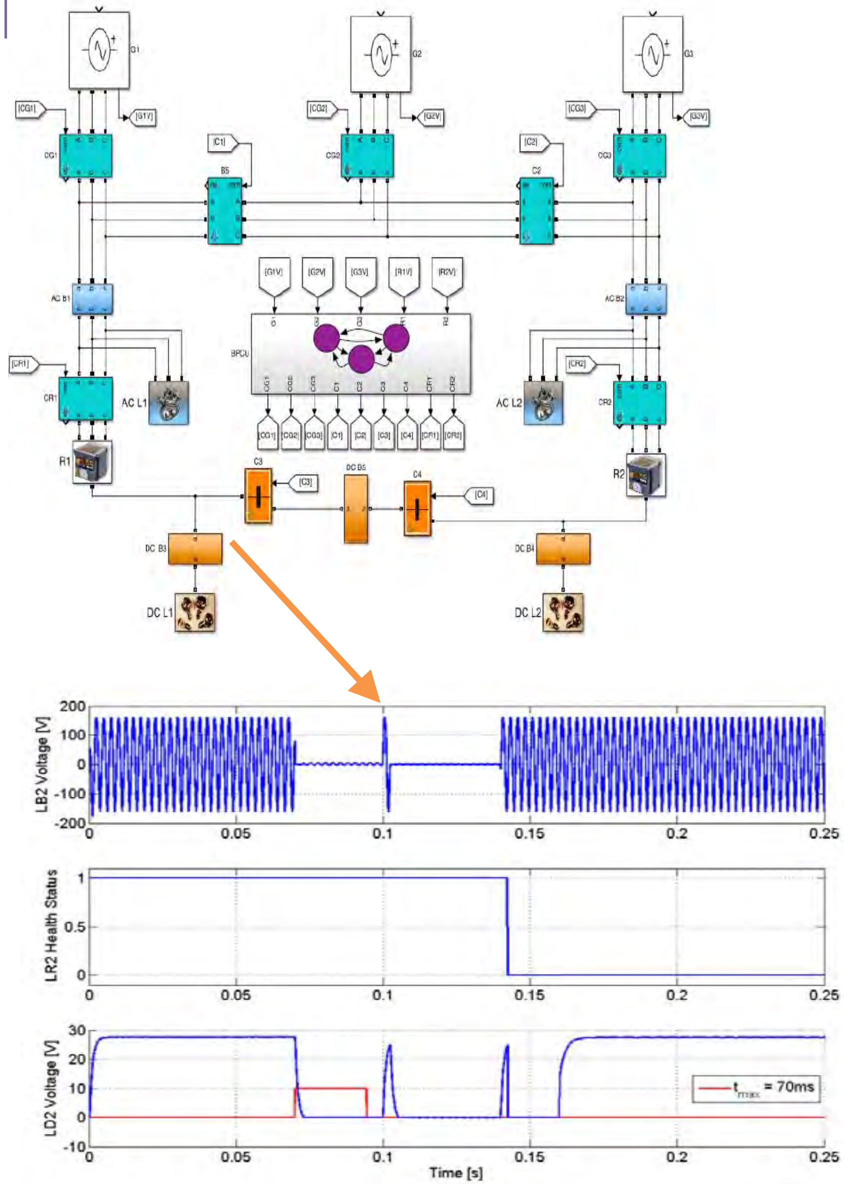
$$\mathcal{E}_S = \{\mathbf{e}_{\mathcal{I}'} | \mathcal{I}' \in h(r_S)\}$$

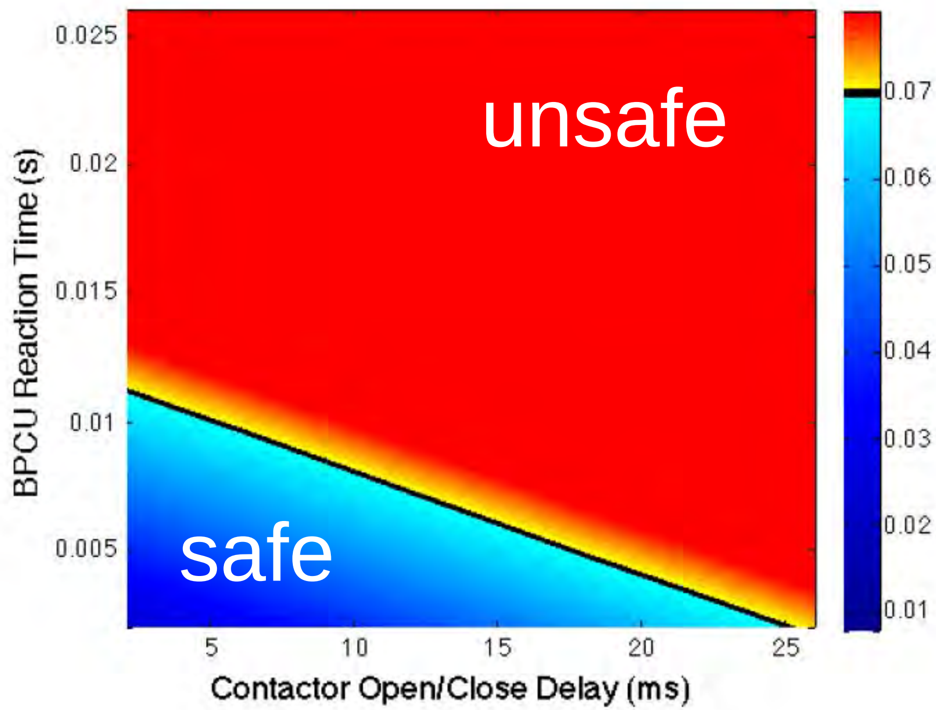
Architecture



- Centralized and distributed controllers for topologies with up to 20 nodes and 20 edges designed using reactive synthesis from Linear Temporal Logic: 4-113 states, 0.5-2 s [TuLiP toolbox]
- Can also support optimal control approaches, e.g., Model Predictive Control

Methodology and Tools: Control Design



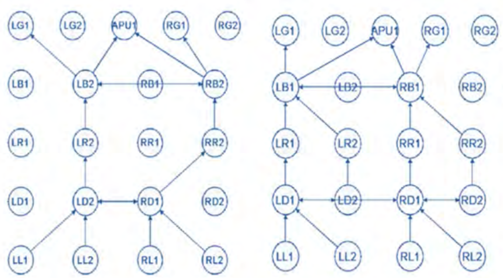


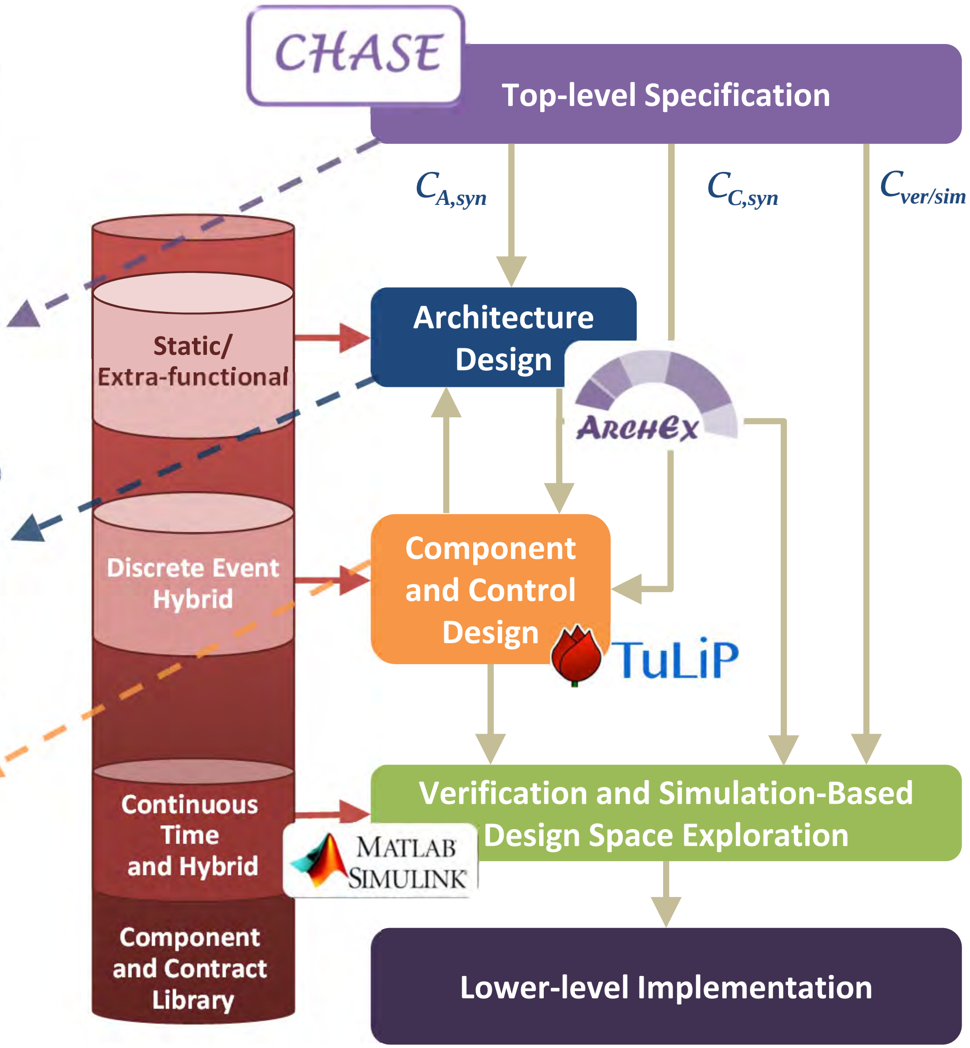
Design Space Exploration: Controller reaction times and contactor delays in the blue region satisfy the requirement [~4 h for a 13x13 point grid]

Verification: Timing violation at the DC bus due to a two- generator fault followed by a rectifier fault (worst case scenario)

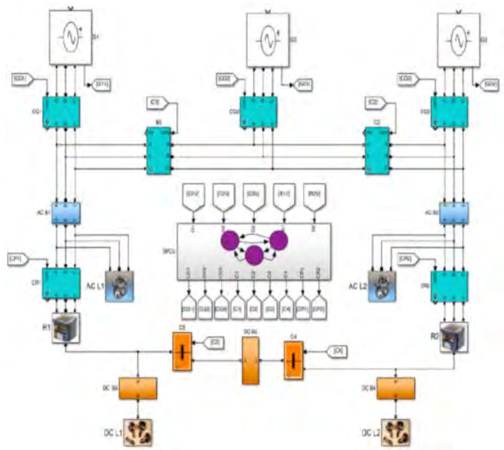
Methodology and Tools: Summary

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

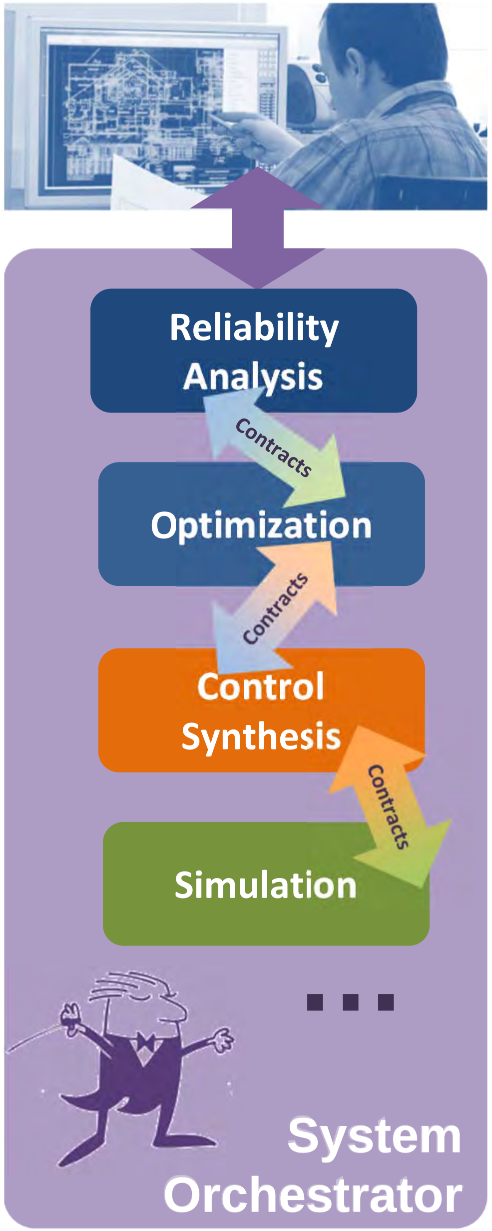




1. No AC bus shall be simultaneously powered by more than one AC source.
2. The aircraft electric power system shall provide power with the following characteristics: 115 +/- 5 V (amplitude) and 400 Hz (frequency) for AC loads and 28 +/-2 V for DC loads.
3. The failure probability at an essential load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



Summary of Contributions



- Developed a **cyber-physical system engineering framework** using a platform-based methodology with assume-guarantee contracts to improve design quality, increase productivity, and reduce costs

- Developed a **contract-based** approach as a foundation encompassing both horizontal and vertical integration steps
  - Enable requirement validation and concurrent development of architectures and embedded control algorithms
  - Formalize refinement between heterogeneous models using vertical contracts

- Developed optimization-based **mapping algorithms** combining approximation and customized solvers for efficient design space exploration of large, mixed discrete-continuous design spaces

- Demonstrated methodology, contract framework, and algorithms on **industrial designs** and **transferred technology** to industry

Moving Forward: Expressive Formalisms, Scalable Algorithms, and "Big Data"

Foundations: Need A/G contracts for "richer" specification formalisms

- Support modular design for **hybrid systems**
- Support modular design under uncertainties (**stochastic** contracts)

Algorithms: Need algorithms that reason about combinations of heterogeneous constraints for scalable analysis and synthesis

- Support more **design concerns**, e.g., security and privacy
- Support more **application domains**: smart grids, autonomous systems, swarm systems, smart cities, …

Data-Driven Design: Closing the loop with data…

- **Design Time**: Use **design data** (e.g., constraint violation) and **operational data** to enrich, validate, and refine components, contracts, and requirements
- **Run Time**: Support modeling, analysis, and design of **learning**-based systems

Questions?