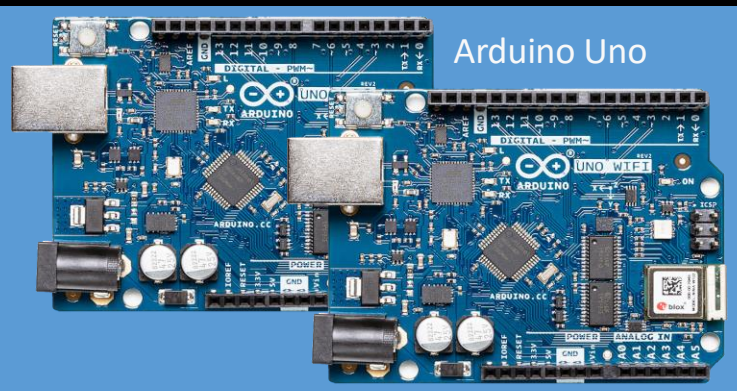
CPS Modeling

Progress and Lessons Learned - CPS 1.0

Janos Sztipanovits
Institute for Convergent Software Integrated Systems
Vanderbilt University

20th Anniversary of the NSF CPS Program
Arlington, VA
October 21, 2026

Embedded Systems (1998 – 2008)



Embedded Computing Device

Sensors

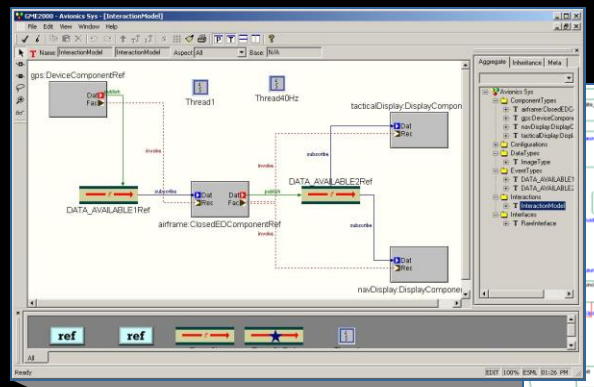
Actuators



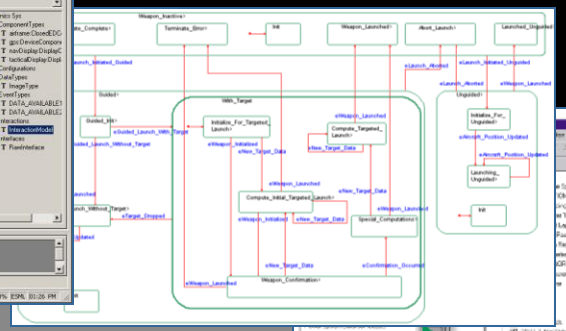
Models of embedded computing device

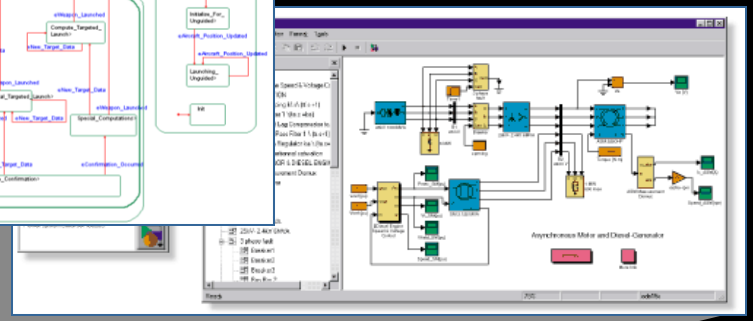
- HW/SW architecture
- Conceptual organization
- Concurrency models
- Timed behavior models
- Performance, scalability
- Power

Model-based Perspective Embedded Systems



Domain-Specific Models





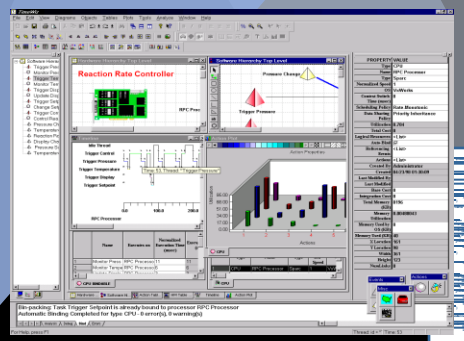
Model Transformations

Model Transformations

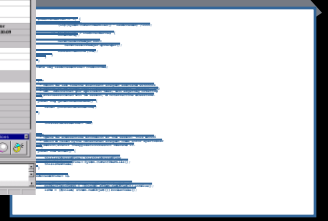
- Code generators
- Multi-target generators
- Formal models of model transformations
- Embeddable generators

```
<CONFIGURATION>
  <PROCESSOR>
    <NAME>OCP_P1</NAME>
    <CONFIGURATION_PASS>
      <HOME>
        <HOME_TYPE> BM_OPEN_ED_COMPONENT </HOME_TYPE>
        <ID>
          <GROUP_ID> 20 </GROUP_ID>
          <ITEM_ID> 22 </ITEM_ID>
          <NAME> BM_OPEN_ED_COMPONENT </NAME>
        </ID>
        <COMPONENT>
          <ID>
            <GROUP_ID>200</GROUP_ID>
            <ITEM_ID>220</ITEM_ID>
            <NAME>WAYPOINT_PROXY</NAME>
          </ID>
          <DISTRIBUTION>
            <DIST_WRITE>
              <PROXY>
                <ID>
                  <GROUP_ID>200</GROUP_ID>
                  <ITEM_ID>221</ITEM_ID>
                  <NAME>WAYPOINT</NAME>
                </ID>
              </PROXY>
            </DIST_WRITE>
          </DISTRIBUTION>
          <EVENT_SUPPLIER>
            <EVENT_SET>
              <EVENT>
```

Configuration Specification



Verification Tools



Code

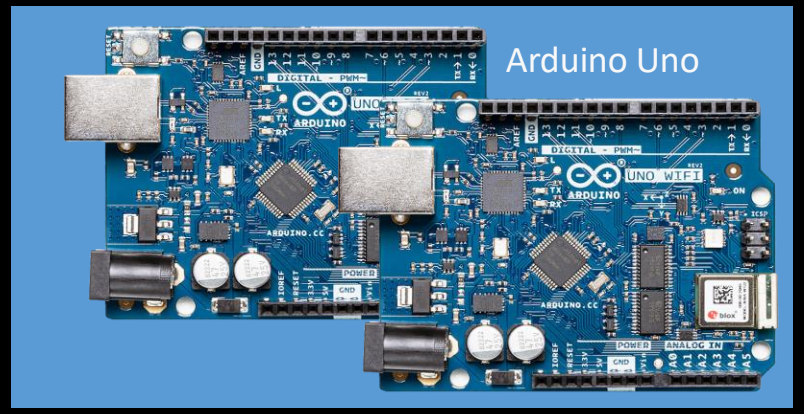
Special Issue on Modeling and Design of Embedded Software
Proceedings of the IEEE, Vol. 91, No.1., January, 2003

Advancements Between 1998-2008

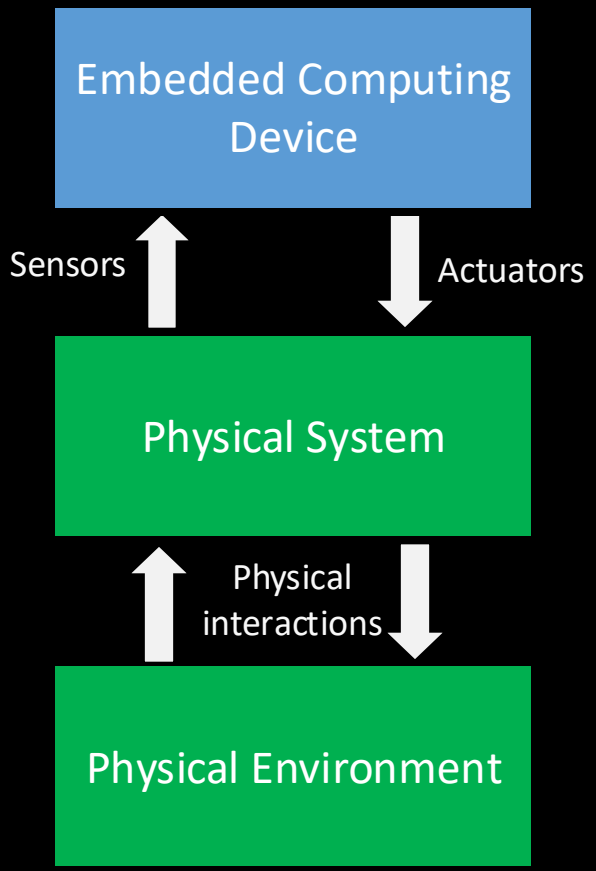
Embedded systems development has undergone a transformative evolution, shifting from code-based design to model-based methods that have become widely adopted in both academia and industry.

1. **Scientific Foundations**
   - Metamodeling
   - Heterogeneous Models of Computation
   - Hybrid Systems
2. **Modeling Languages**
   - Simulink/Stateflow
   - Architecture Analysis and Design Lng (AADL)
   - SySML…
3. **Tools**
   - Eclipse Modeling Framework
   - Matlab tool suite
   - Ptolemy II (UC Berkeley)
   - Generic Modeling Environment (Vanderbilt)
4. **Industry Adoption**
   - Automotive (BMW, Ford, GM,..)
   - Aerospace (Boeing, Airbus, Honeywell..)
   - Electronics (SLD tools)
5. **Major Research Initiatives**
   - DARPA's SEC, PCES, MoBIES, NEST, ARMS
   - NSF ITR program
   - EU IST - ARTIST projects
6. **Technology Contributions**
   - Hybrid systems tools (HyTech, PHAVer..)
   - Composition platforms (TTA, BIP..)
   - Platform-based design (Metropolis..)
   - Verification tools (UPPAAL, SPIN..)

Moving Beyond Embedded Systems to CPS (2006 - .....)







Models of embedded computing device

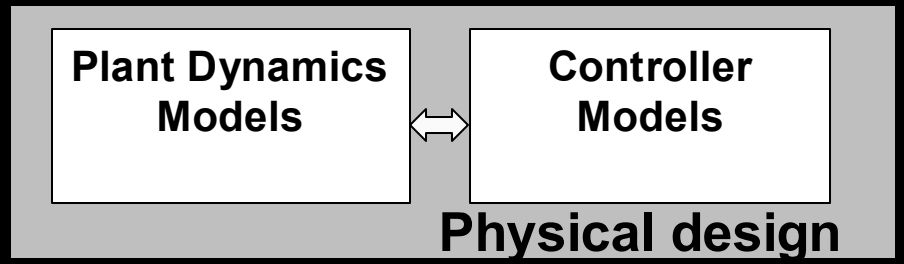
- HW/SW architecture
- Conceptual organization
- Concurrency models
- Timed behavior models
- Performance, scalability
- Power

Physical Models

- Physical architecture
- Dynamics
- Faults and their effects
- Aleatoric and epistemic uncertainties
- Energy flows

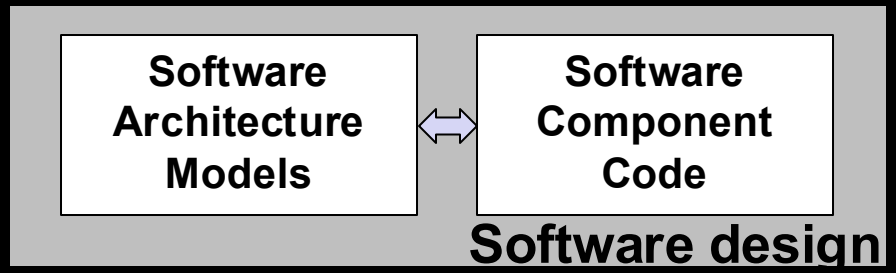
- Embedded Systems: focus on impacts of physical environments on computing
- Cyber-Physical Systems (CPS) look both sides together

CPS Design Progresses in Abstraction Layers



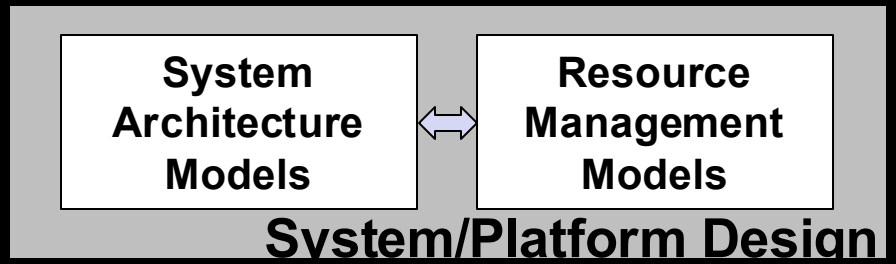
Dynamics: $B(t) = \kappa_p(B_1(t),...,B_j(t))$

- *Properties*: stability, safety, performance
- *Abstractions*: continuous time, functions, signals, flows,…



Software: $B(i) = \kappa_c(B_1(i),...,B_k(i))$

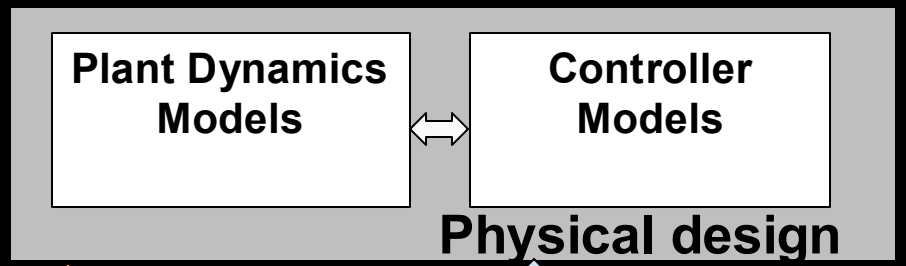
- *Properties*: state invariants, time limits,..
- *Abstractions*: logical-time, concurrency, atomicity,,..



Platform: $B(t_j) = \kappa_p(B_1(t_i),...,B_k(t_i))$

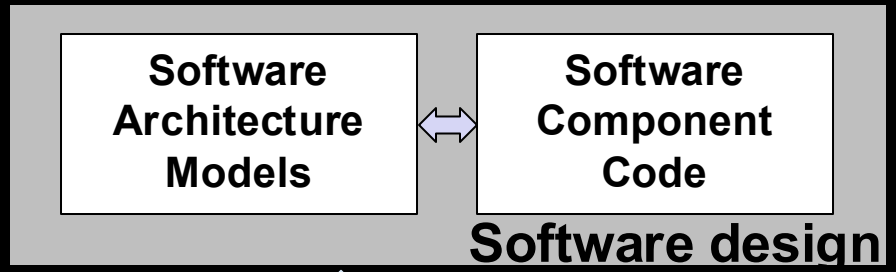
- *Properties*: timing, power, security, fault tolerance
- *Abstractions*: discrete-time, delays, resources, scheduling,

Problems With Leaky Abstractions



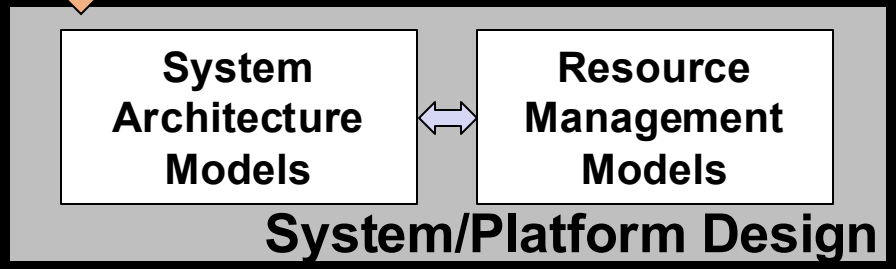
Controller dynamics is developed without considering implementation uncertainties (e.g. word length, clock accuracy) while optimizing performance.

Assumption: Effects of digital implementation can be neglected



***Software** is* developed without explicitly considering systems platform characteristics, even though key behavioral properties depend on it.

Assumption: Effects of platform properties can be neglected



Platform design creates side effects such as scheduling jitters and network uncertainties, roundoff errors etc. that may require re-verification of key properties on all levels.

CPS Enabled Co-Design Approaches

- Leaky abstractions cause loss of compositionality across system layers.
  Consequences:
  - intractable interactions
  - unpredictable system level behavior
  - full-system verification does not scale
- CPS solutions:
  - ***Robustness:*** Use design constraints that decouple systems layers for selected properties. Example: Passivity-based design makes stability robust against implementation side effects
  - ***Cross-layer modeling:*** Develop methods for modeling cross-layer interactions

J. Sztipanovits*; X. Koutsoukos; G. Karsai; N. Kottenstette; P. Antsaklis; V. Gupta*., "Toward a Science of Cyber–Physical System Integration," in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 29-44, Jan. 2012

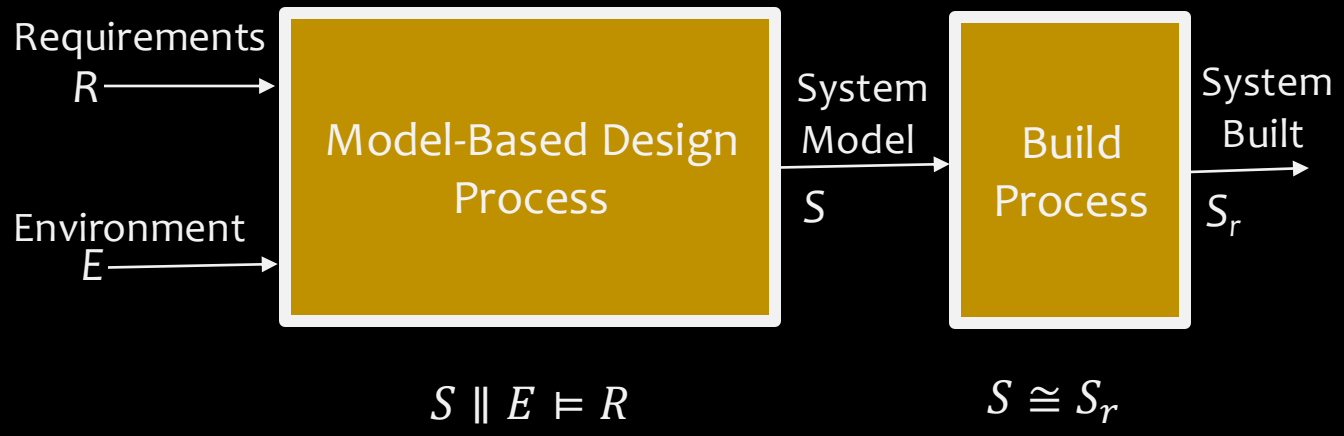
Cyber-Physical Systems (2008 – Present)

System Types	Characteristics	Challenges	
Individual System	• Functions emerge from the interaction of physical and computational processes • Robots, vehicles, industrial plants,…	• Safety • Security • Assurance	
Human Cyber-Physical System (HCPS)	• CPS integrated with human(s) • Humans are integral part of dynamics • Human/AI/Machine teams	• Modeling human agents • Human/Machine Interface • Mutual introspection	
Societal-Scale Human Cyber-Physical System	• Heterogeneous collections of interacting systems • Smart cities, communities, infrastructures • Humans are part of control loops	• Mixed autonomy • Incentive engineering and mechanism design • Social context • Responsible computing	

Scale-up Challenge for Model-based Design

Large scale test – DARPA Adaptive Vehicle Make (AVM) program: (2010-2014)

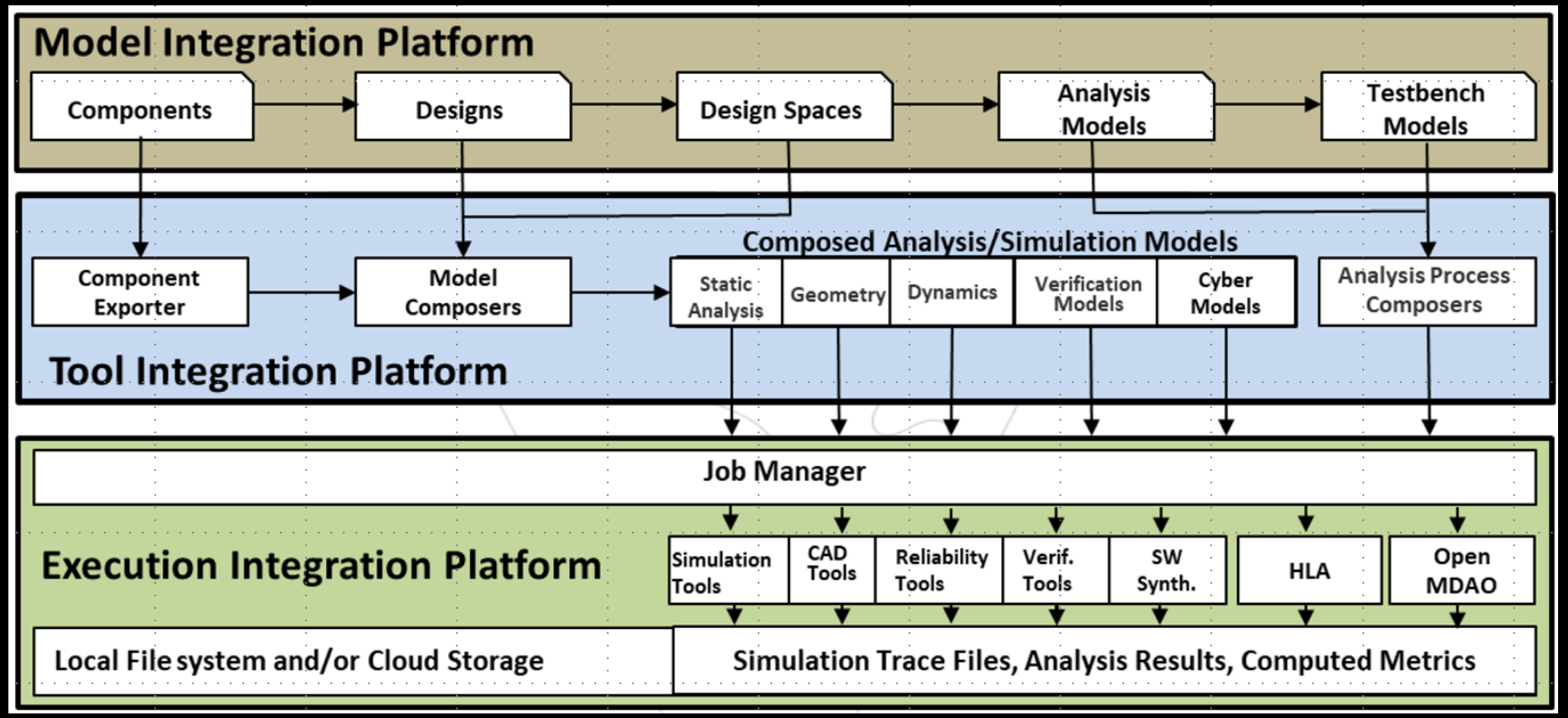
- **CPS "Design House" + foundry-style manufacturing**
- Cloud deployed **design automation tool suite** (OpenMETA) supporting crowed-sourcing
- End-to-end design **incorporating manufacturability**
- **Heavy amphibious vehicle designed and manufactured**



How to move toward correct-by-construction?

- Available tool suites follow separation of concerns – impact is a concern…
- We need cross-domain modeling
- Vertically integrated tool suites should be complemented by **horizontal integration platforms**
- Integration platforms should extend to **modeling**, **analysis** and **verification** and **execution**

OpenMETA Horizontal Integration Platforms

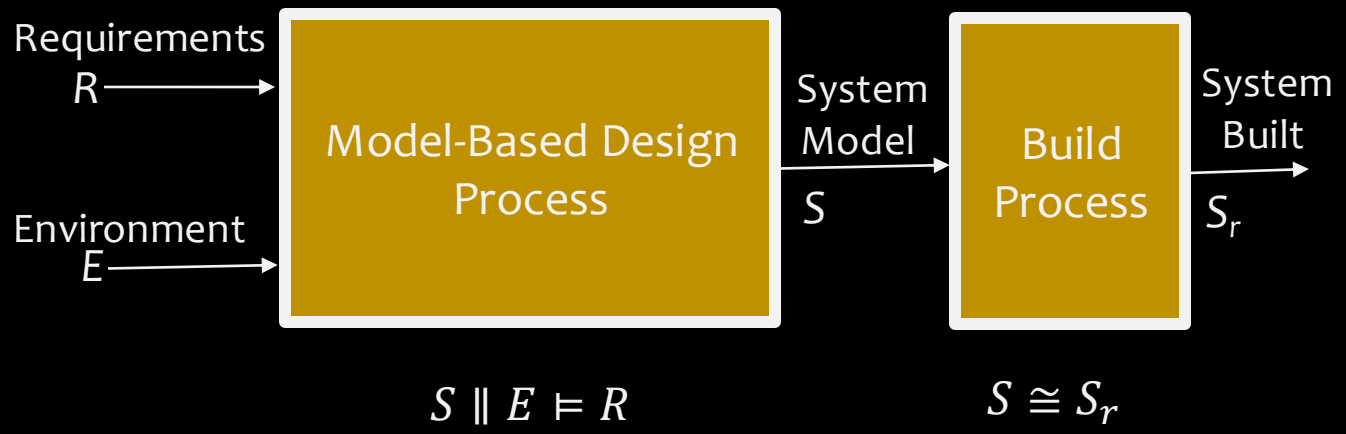


Horizontal Integration Platforms cut across traditionally isolated design domains.

Is Model-based Design Meets CPS Challenges?

Large scale test – DARPA Adaptive Vehicle Make (AVM) program: (2010-2014)

- **CPS "Design House" + foundry-style manufacturing**
- Cloud deployed **design automation tool suite** (OpenMETA) supporting crowed-sourcing
- End-to-end design **incorporating manufacturability**
- **Heavy amphibious vehicle designed and manufactured**



LIMITS

- Creating multi-fidelity model libraries is hard and **very expensive**
- **Limited scalability** of analysis tools (e.g. simulators) and verification tools
- Impact of **epistemic uncertainty** in physical component modeling
- Limited scope of design-space construction and scalability limits of design-space exploration
- Cost of precise modeling of manufacturing side-effects

J. Sztipanovits, Ted Bapty, E. Jackson, X. Koutsoukos, Zs. Lattman, S. Neema: "Model and Tool Integration Platform for Cyber-Physical System Design", *Proceedings of the IEEE*, special issue on Design Automation for Cyber-Physical Systems, 2018. DOI: 10.1109/JPROC.2018.2838530

Integrating Model-based and Data-driven approaches driven by AI is the story of CPS – 2?

CPS as BIG IDEA:
Convergence Among Physical and Computational Sciences

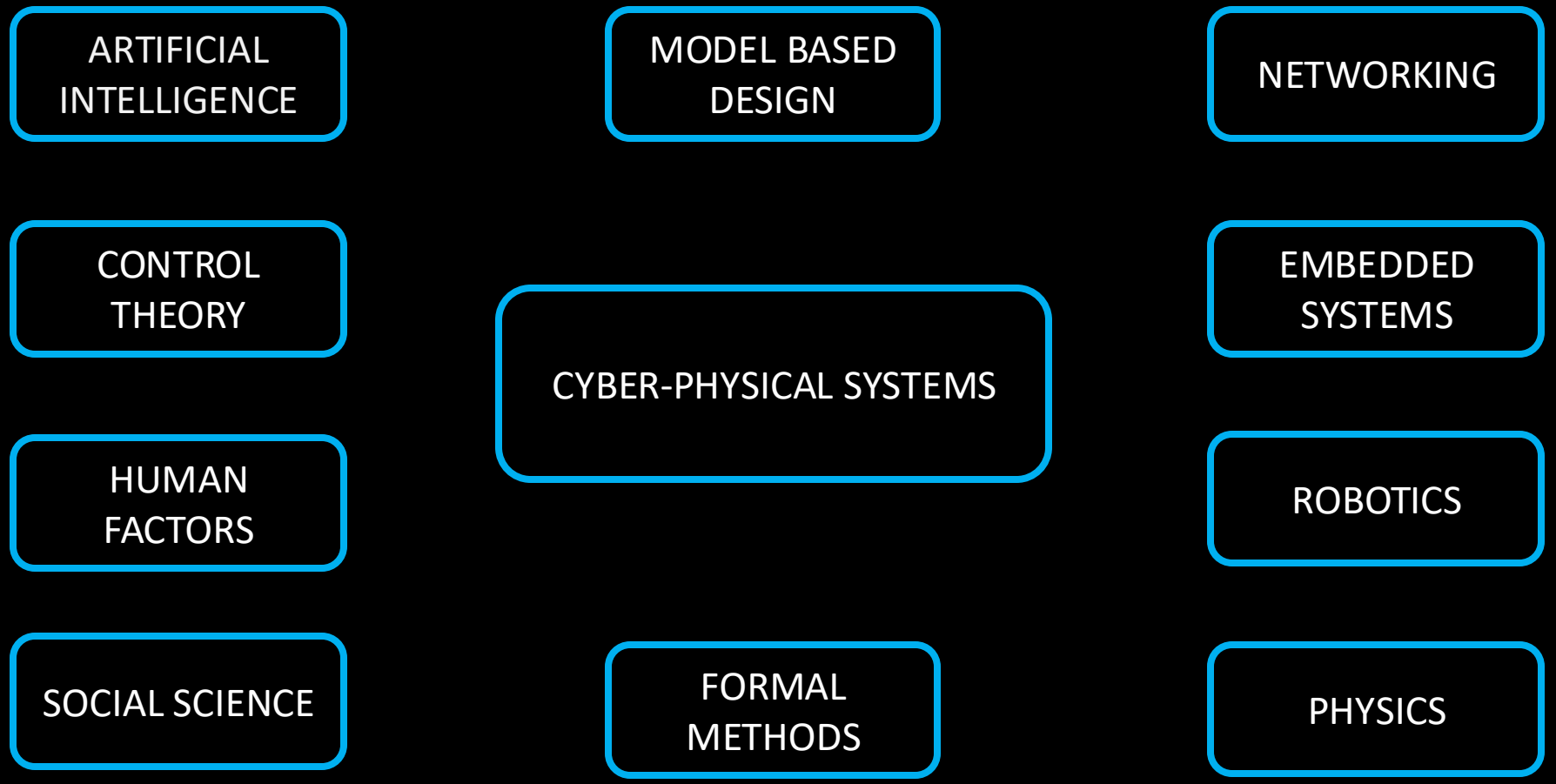
We see alternating periods of disciplinary specialization and cross-disciplinary synthesis, often triggered by scientific revolutions or technological needs.

- ~1980–2010: Disciplinary Depth with Synthetic Trends
  - Disciplinary Depth: Particle physics (e.g., Large Hadron Collider), genomics, and computer science subfields
  - Synthesis: Interdisciplinary fields like bioinformatics, nanotechnology, and environmental science

- ~2010–Present: Synthesis Dominating
  - One of NSF's 10 Big Ideas – 2016: Growing Convergence Research
  - Artificial intelligence integrates computer science, neuroscience, and mathematics
  - Synthetic biology combines biology, engineering, chemistry and computer science
  - Climate science blends atmospheric physics, chemistry, economics, and policy

Why Synthesis?: **Wicked Problems (e.g., pandemics, AI ethics)**
Advances in computing (e.g., Internet, collaborative platform, AI)

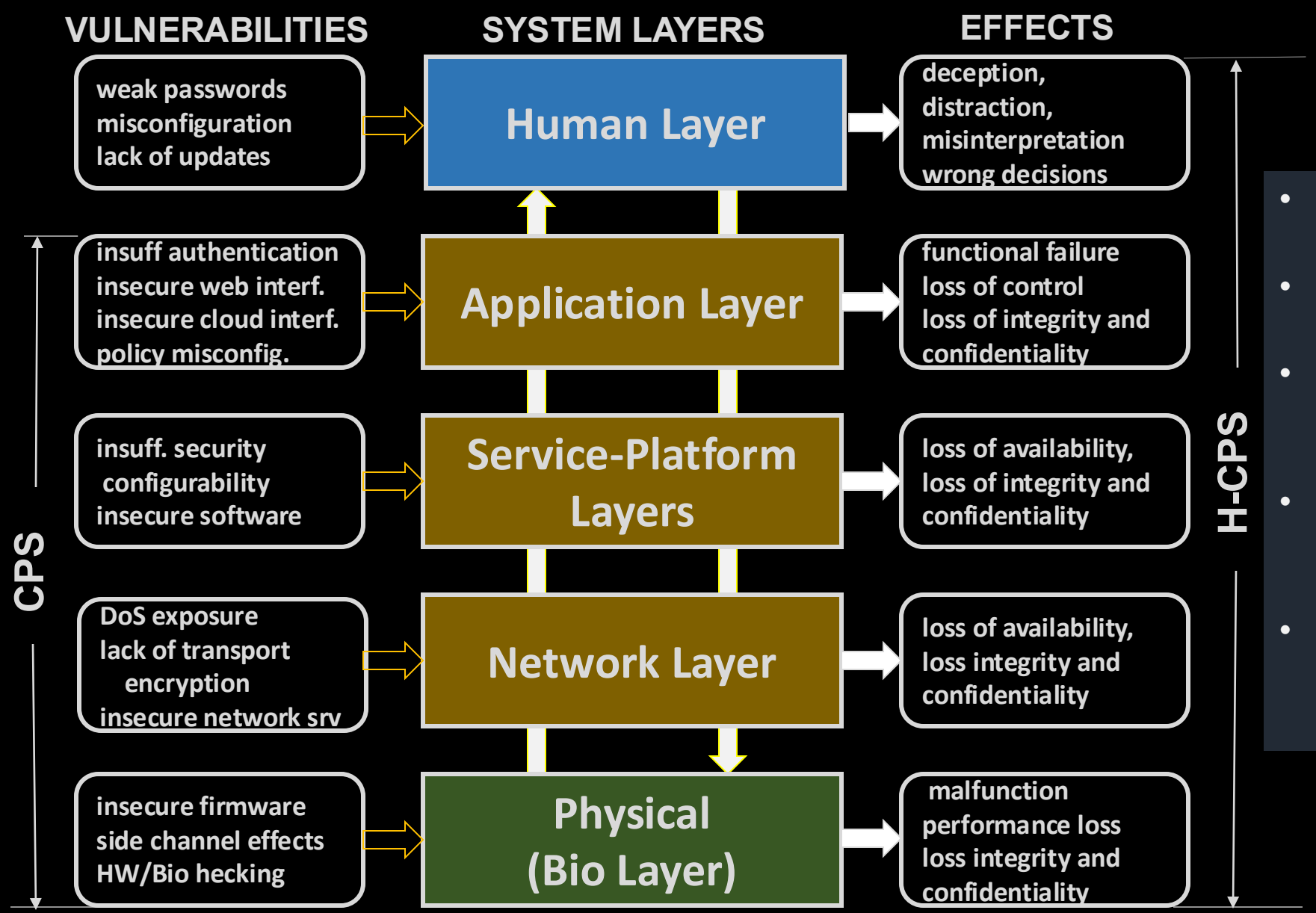
CPS Is the Ultimate Example for Impacts of Convergence – Now Accelerated by AI



Convergence: great benefits, exciting future

What is the price?

Security Challenge in CPS



- Effects propagate across layers
- Easy to lose tractability
- Full isolation of layers, components is hard
- Long supply chains and hard to trace provenance
- Coordinated physical/bio and cyber attacks are extremely hard to detect/prevent

CPS Security is a Real Problem!

CLAROTY Global Survey Report on CPS Security in 2024

1. Incidents reported by over 1,100 leads
   - 45% reported financial impact over $500,000
   - 17% reported loss of over $1 million
2. Ransomware
   - 53% met ransom demands of more than $500,000
   - 78% reported ransom payments over $500,000 in healthcare sector
3. Critical Infrastructure Security…

The benefits we reap and the price we pay

It's easier to find payer for immediate benefits
than for preventing future harm.

But one way or another – the costs will eventually come due