Introduction to Civil Information Systems

Mark A. Austin

University of Maryland

austin@umd.edu

ENCE 688R, Spring Semester 2023

January 27, 2023

Overview

Part 4

Cyber-Physical Systems

New Computing Infrastructure $\rightarrow$ New System Abstractions

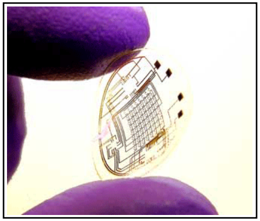
Cyber-Physical Systems

General Idea

Embedded computers and networks monitor and control the physical processes, usually with feedback loops where computation affects physical processes, and vice versa.

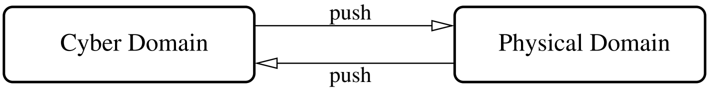
Two Examples

Programmable Contact Lens



Programmable Windows

Cyber-Physical Systems Overview



C–P Structure

Cyber capability in every physical component	Spatial and network abstractions
Executable code	–– physical spaces
Networks of computation	–– networks of networks
Heterogeneous implementations	Sensors and actuators.

C–P Behavior

Dominated by logic	Physics from multiple domains.
Control, communications	Combined logic and differential equations.
Stringent requirements on timing	Not entirely predictable.
Needs to be fault tolerant	Multiple spatial– and temporal– resolutions.

Cyber-Physical Systems

Physical System Concerns

- Design success corresponds to notions of resilience and reliability.
- Behavior is constrained by conservation laws (e.g., conservation of mass, conservation of momentum, conservation of energy, etc..).
- Behavior often described by families of differential equations.
- Behavior tends to be continuous – usually there will be warning of imminent failure.
- Behavior may not be deterministic – this aspect of physical systems leads to the need for reliability analysis.
- For design purposes, uncertainties in behavior are often handled through the use of safety factors.

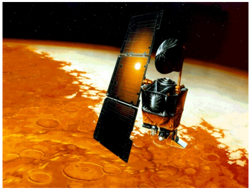
Cyber-Physical Systems

Software System Concerns

- Design success corresponds to notions of correctness of functionality and timeliness of computation.
- Computational systems are discrete and inherently logical. Notions of energy conservation ...etc... and differential equations do not apply.
- Does not make sense to apply a safety factor. If a computational strategy is logically incorrect, then “saying it louder” will not fix anything.
- The main benefit of software is that functionality can be programmed and then re-programmed at a later date.
- A small logical error can result in a system-wide failure.

Cyber-Physical Systems (Notable Failures)

Example 1. NASA's Mars Climate Orbiter, September 1999.



NASA's systems engineering process did not specify the system of measurement. One of the development teams used Imperial measurement; the other metric.

When parameters from one module were passed to another during orbit navigation correct, no conversion was performed, resulting in $125m loss.

Cyber-Physical Systems (Notable Failures)

Example 2. Denver Airport Baggage Handling System

1995. Baggage handling system is 26 miles of conveyors; 300 computers. Fixing the incredibly buggy system requires additional 50 percent of the original budget - nearly $200m.

2005. System still does not work. Airport managers revert to baggage carts with human drivers.

Source: Jackson, Scientific American, June 2006.

Cyber-Physical Systems (Error-Free Software)

Embedded computer systems and software need to deliver functionality that is correct and works with no errors.

CPS Design Requirements:

- Reactivity: System response need to occur within a known bounded range and delay.
- Autonomy: Systems need to provide continuous service without human intervention.
- Dependability: Systems need to be resilient to attack and hardware/software failures.
- Scaleability: System performance needs to scale with supplied resources.

Software for smart electronic devices is how Java got started !!!

Causes of Software-Related Accidents

Modern Software

Modern software is simply the design of a machine abstracted from its physical realization.

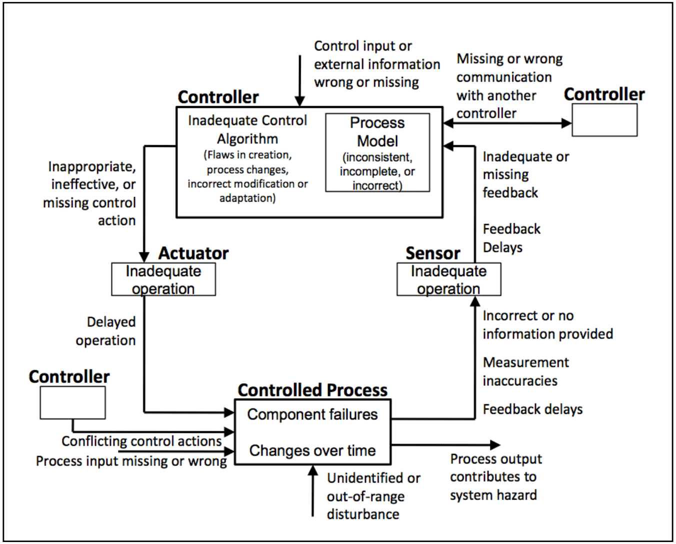
Software Accidents

Software accidents are usually caused by flawed requirements and not standard wear-out failures.

This includes:

- Incomplete (or wrong) assumptions about the operation of the controlled system or required operation of the software.
- Unhandled control system states and environmental conditions.

Engineering Sensor Systems (Error-Free Software)

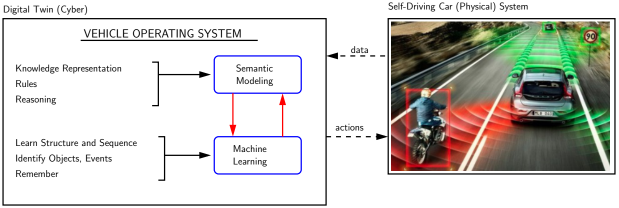

Control input or external information wrong or missing
Missing or wrong communication with another controller
Controller
Controller
Inadequate Control Algorithm (Flaws in creation, process changes, incorrect modification or adaptation)
Process Model (inconsistent, incomplete, or incorrect)
Inappropriate, ineffective, or missing control action
Inadequate or missing feedback
Feedback Delays
Actuator
Inadequate operation
Sensor
Inadequate operation
Delayed operation
Incorrect or no information provided
Measurement inaccuracies
Feedback delays
Controller
Controlled Process
Component failures
Changes over time
Conflicting control actions
Process input missing or wrong
Unidentified or out-of-range disturbance
Process output contributes to system hazard

Digital Twin Systems

New Computing Infrastructure $\rightarrow$ New System Abstractions

Digital Twins (2000-today)

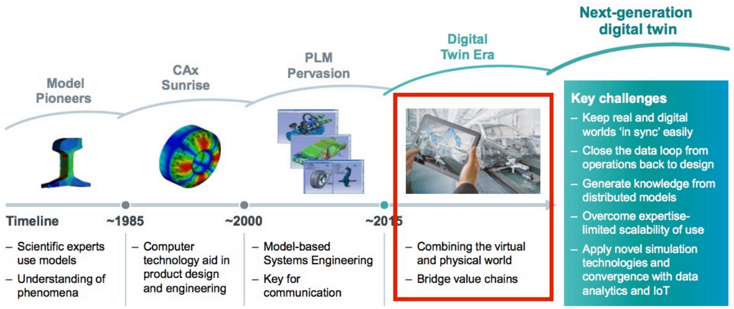
Definition. Virtual representation of a physical object or system that operates across the system lifecycle (not just the front end).



Required Functionality

- Mirror implementation of physical world through real-time monitoring and synchronization of data with events.
- Provide algorithms and software for observation, reasoning, and physical systems control.

Digital Twins (Business Case + Applications)


Model Pioneers
CAx Sunrise
PLM Pervasion
Digital Twin Era
Next-generation digital twin
Timeline
~1985
~2000
~2015
– Scientific experts use models
– Understanding of phenomena
– Computer technology aid in product design and engineering
– Model-based Systems Engineering
– Key for communication
– Combining the virtual and physical world
– Bridge value chains
Key challenges
– Keep real and digital worlds 'in sync' easily
– Close the data loop from operations back to design
– Generate knowledge from distributed models
– Overcome expertise-limited scalability of use
– Apply novel simulation technologies and convergence with data analytics and IoT

Many Applications

- NASA Spacecraft
- Manufacturing processes
- Building operations
- Personalized medicine
- Smart Cities
- ... etc.

Digital Twins (Technical Implementation)

Technical Implementation (2023, Google, Siemens, IBM)

- AI and ML will be deeply embedded in new software and algorithms.

Artificial Intelligence:

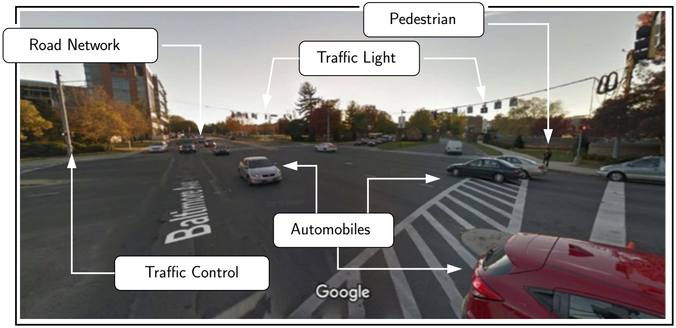
- Knowledge representation and reasoning with ontologies and rules. Semantic graphs. Executable event-based processing.

Machine Learning:

- Modern neural networks. Input-to-output prediction.
- Data mining.
- Identify objects, events, and anomalies.
- Learn structure and sequence. Remember stuff.

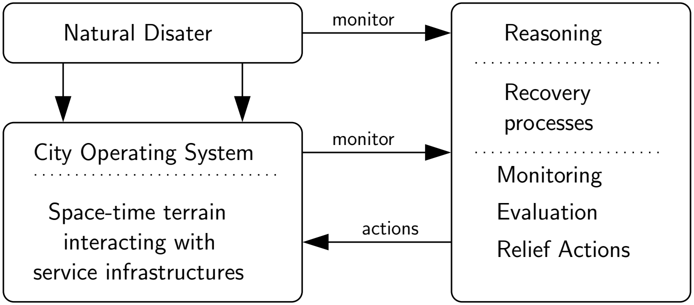
Digital Twin Application (Self-Drivng Car)

Goal. How to traverse traffic intersection safely and without causing an accident?

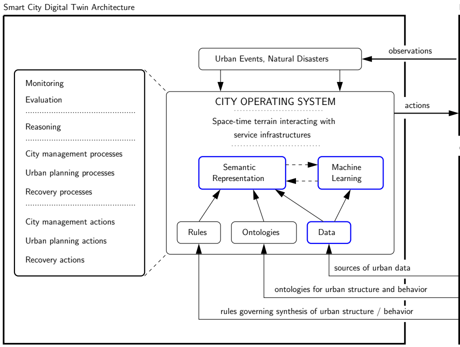


Required Capability. Observe, evaluate, reason, take actions.
Challenges. Multiple domains, multiple streams of heterogeneous data, event-driven behavior, dynamic, time critical.

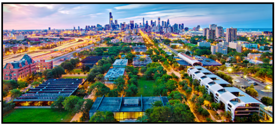
Digital Twin: City Operating Systems



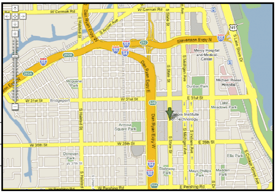
Smart City Digital Twins (2018-2019)



Downtown, Chicago, USA

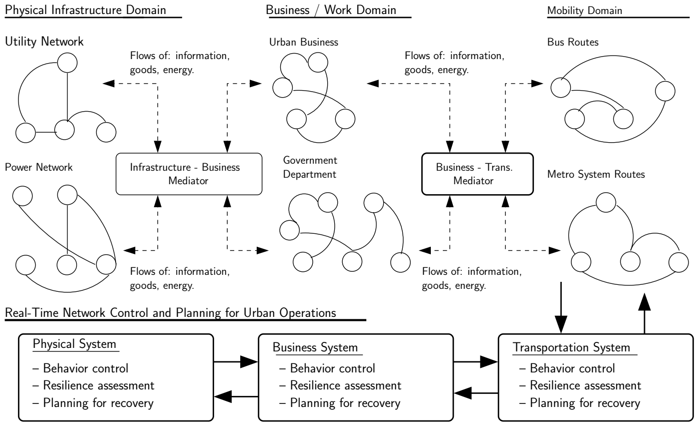


Geo-informatics: Urban data, ontologies and rules.



Required Capability. Monitoring and control of urban processes.
Complications. Potentially, a very large number of digital twins. Distributed decision making.

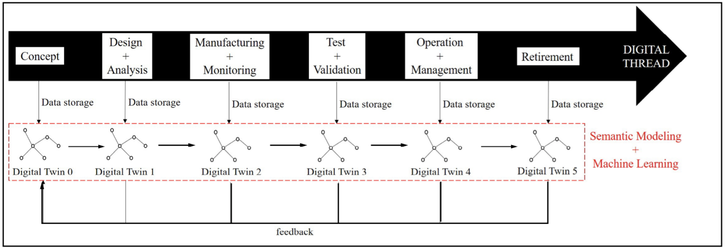
Smart City Digital Twins (2018-2019)



Requirements. Support for digital twin individuals and digital twin communities.

Digital Thread Systems

Digital Threads: (Cradle-to-Grave Lifecycle Support) ...

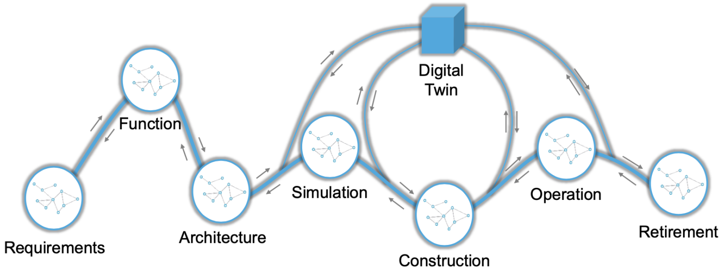


Graph-based Approach

A lot of model-centric engineering boils down to representation of systems as graphs and sequences of graph transformations punctuated by decision making and work/actions.

Digital Thread Systems

Digital Thread System at INL: (Conceptual Model) ...



Def'n: A digital thread is an interconnected software data exchange used to enable digital engineering and digital twinning systems ...

Source: Coelho and Browning, INL, 2022.

References

- Array of Things: See https://arrayofthings.github.io
- Austin M.A., Delgoshaei P., Coelho M. and Heidarinejad M. , Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach, Journal of Management in Engineering, ASCE, Volume 36, Issue 4, July, 2020.
- Bello J.P. et al., SONYC: A System for Monitoring, Analyzing, and Mitigating Urban Noise Pollution, Communications of the ACM, 62, 2, 2019, pp. 68-77.
- Coelho M., and Browning L.S., INL Digital Engineering: Model-Based Design, Digital Threads, Digital Twins, Artificial Intelligence, and Extended Reality for Complex Energy Systems, INL/CON-22-69247, Idaho National Laboratory, Idaho Falls, Idaho 83415, September, 2022.
- Leveson N.G., A New Approach to Software Systems Safety Engineering, System Safety Engineering: Back to the Future, MIT, 2006.
- Tien J.M., Toward a Decision Informatics Paradigm: A Real-Time Information-Based Approach, to Decision Making, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 33, No. 1, February, 2003.