Reimagining Simulation in Discrete Event Logistics Systems

LEON MCGINNIS
PROFESSOR EMERITUS
STEWART SCHOOL OF INDUSTRIAL AND SYSTEMS ENGINEERING
GEORGIA TECH

Four take-aways

- See simulation and digital twin from a broader perspective
- See fundamental challenges of digital twins in our domain
- See potential approaches for resolving those challenges
- See a major roadblock to ultimate success

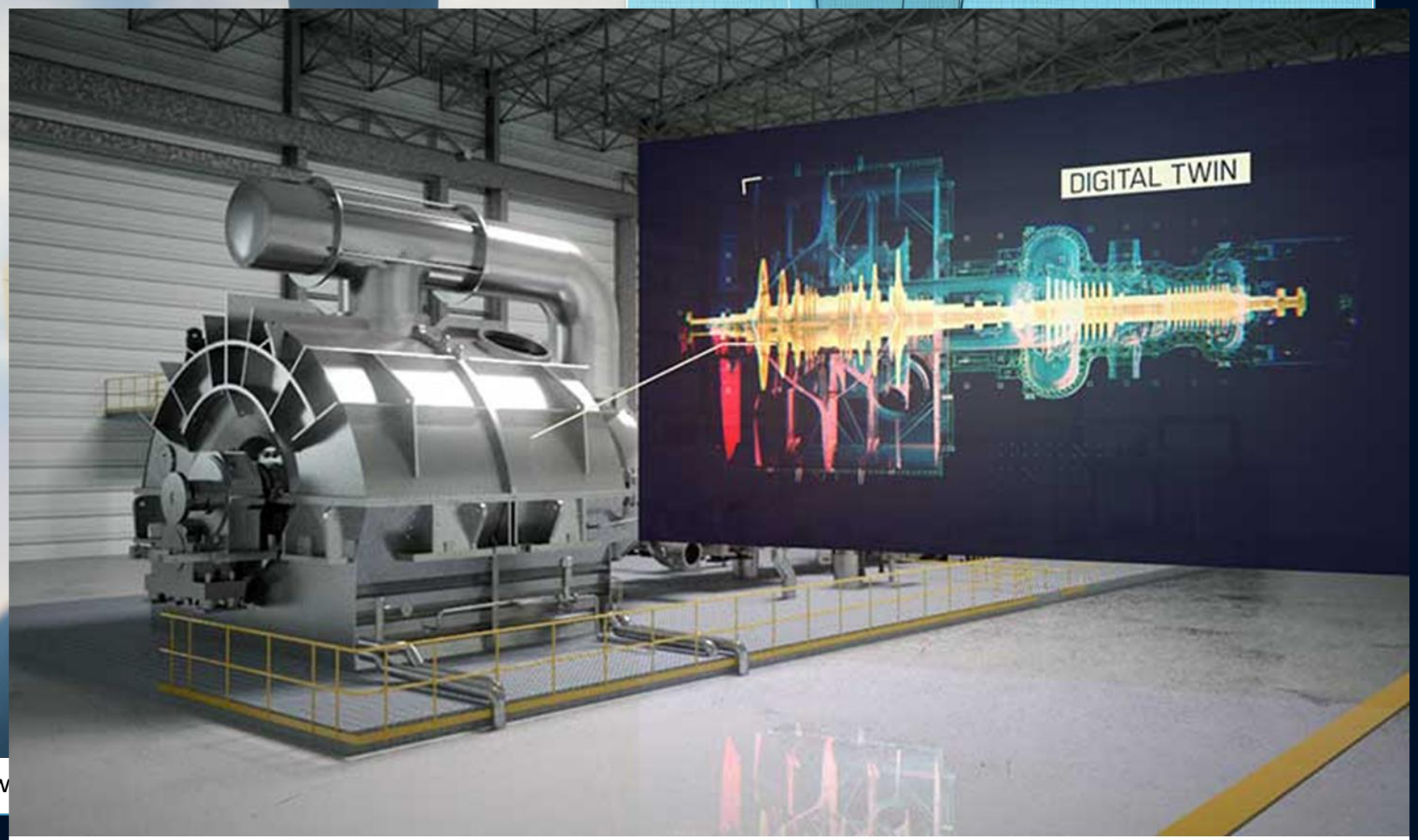
Discrete Event Logistics Systems (DELS)

A network of resources through which objects move and are transformed to a higher value by processes with discrete start and end times. Operations are *managed* to achieve objectives

Objects: material, products, people, information
Resources: machine tools, factories/3PLs, doctors, ICT
Processes: conversion, production/logistics, diagnosis/treatment, computation/presentation, moving, storing

DELS are ubiquitous, often large-scale and complicated, often emergent.

What is a “Digital Twin”?



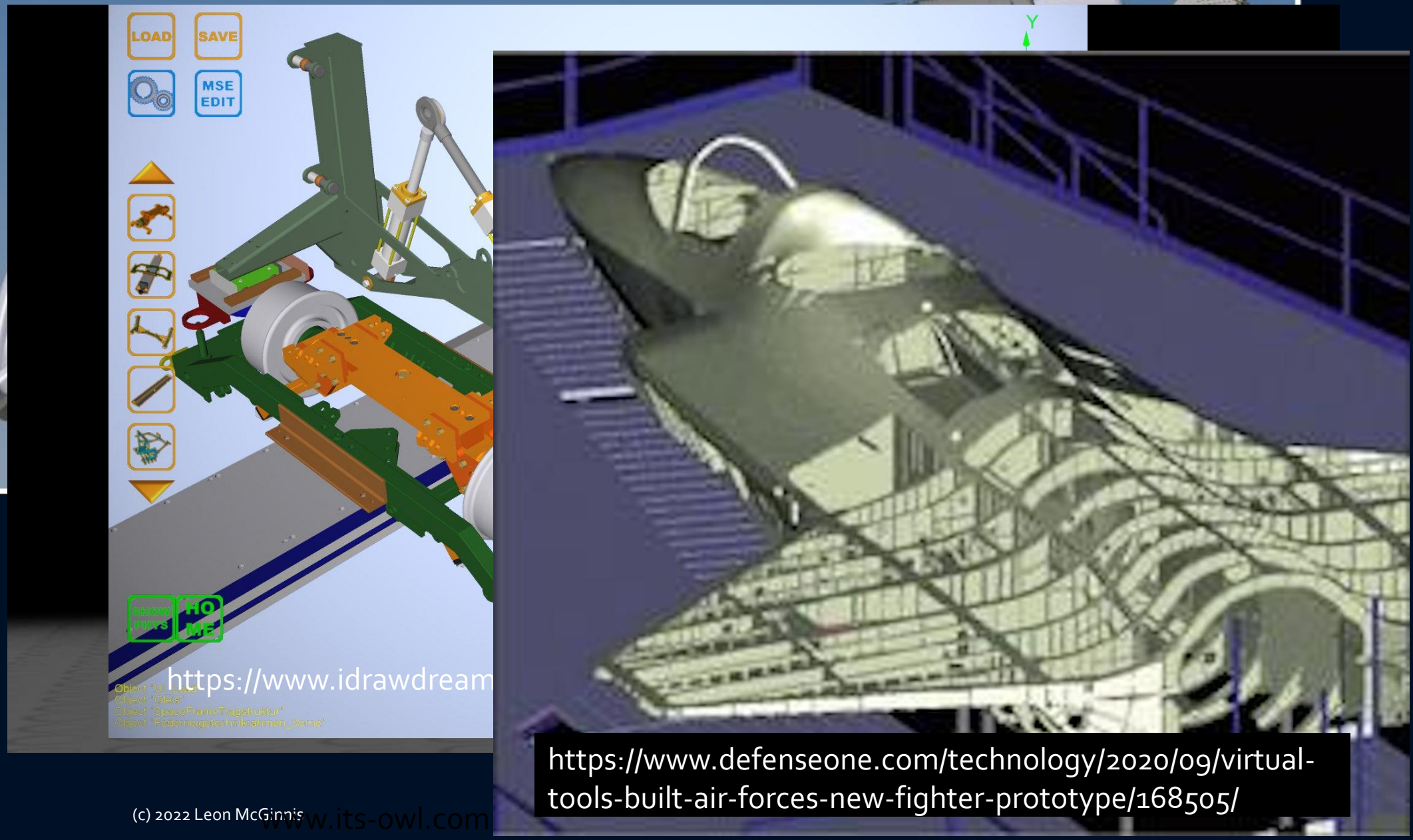
Future Factories: How AI Enables Smart Manufacturing (topbots.com)

Observe DT Applications

- Asset management
  - Monitor state
  - Predict state

Observe DT Enabling Technologies

- Structural (CAD) models
  - Analysis agnostic
- Behavior models
  - Physics-based
  - ***Control theories***
- Visualization
  - Sensors, communication
  - Dashboards

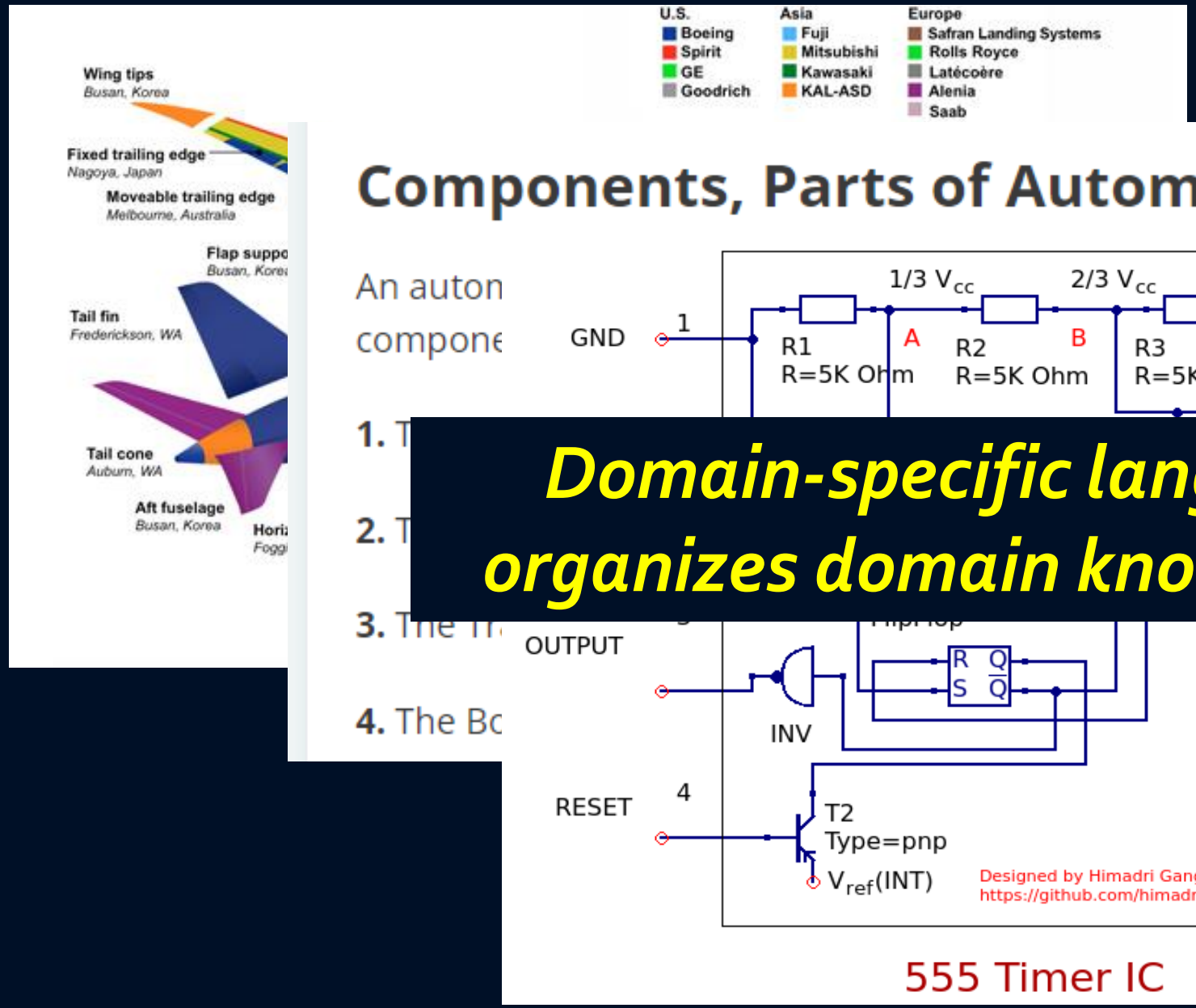


Virtual DT Applications

- Domains
  - Rotating machinery
  - Automobiles
  - Aircraft
  - Integrated circuits

Design-driven Enabling Technologies

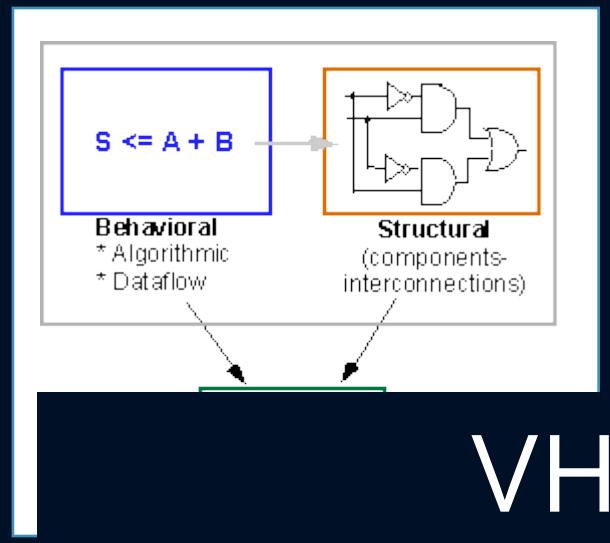
- Domain-specific language
- Analysis integrated with analysis-agnostic system model

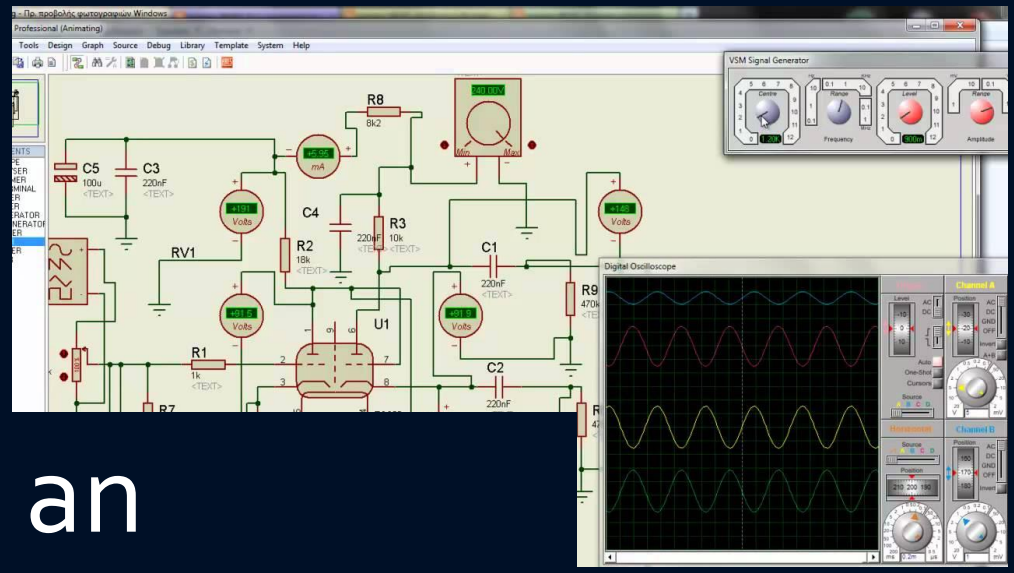


Domain-specific language organizes domain knowledge!

eN3LF.png (1033×627) (imgur.com)

Very High Speed Integrated Circuit Hardware Description Language VHDL

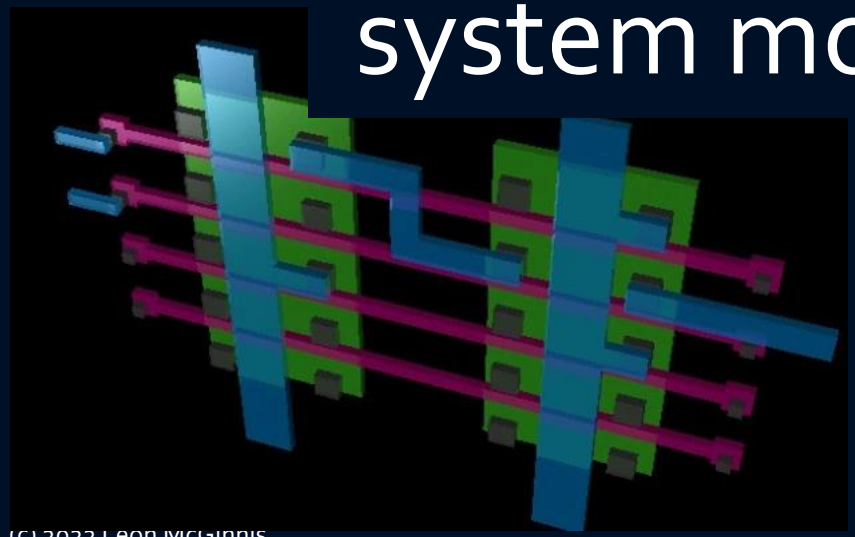


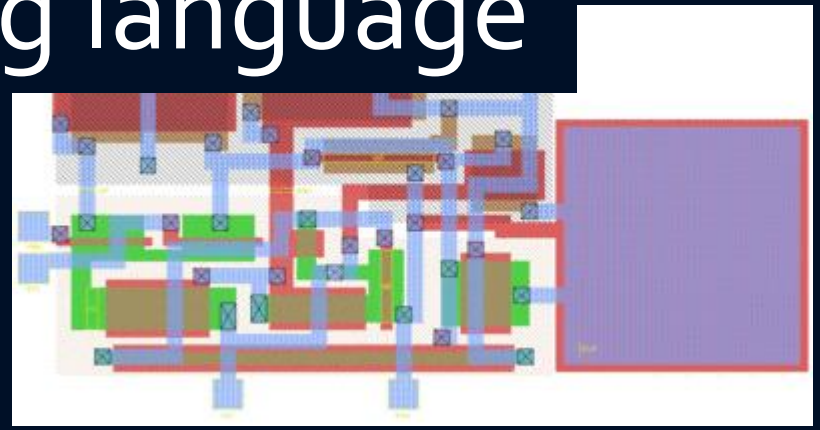


VHDL is an

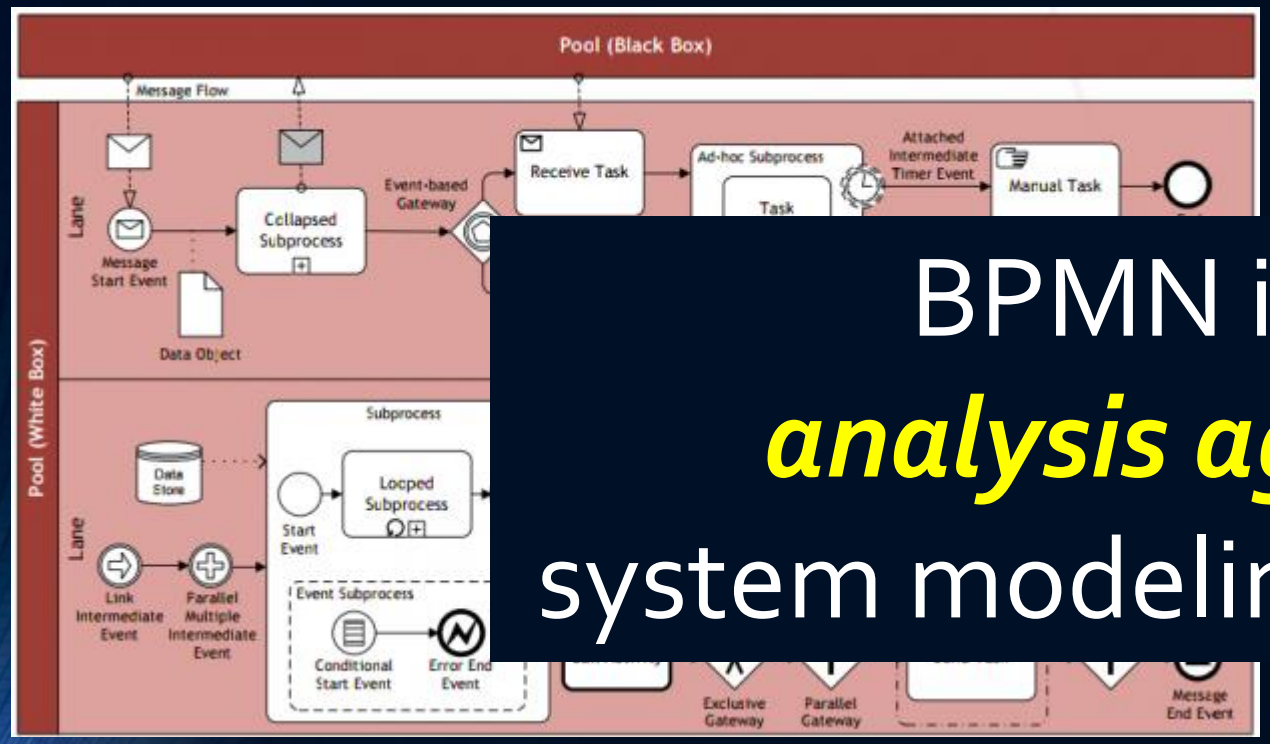
analysis agnostic

system modeling language





BPMN



https://ccmi.fit.cvut.cz/wp-content/uploads/2013/02/bpmn_poster_part-624x364.png

visual modeling language for business analysis applications and specifying enterprise

vs, which is

d notation for

arts that is

usiness

process workflows.

https://www.visual-paradigm.com/guide/bpmn/what-is-bpmn/

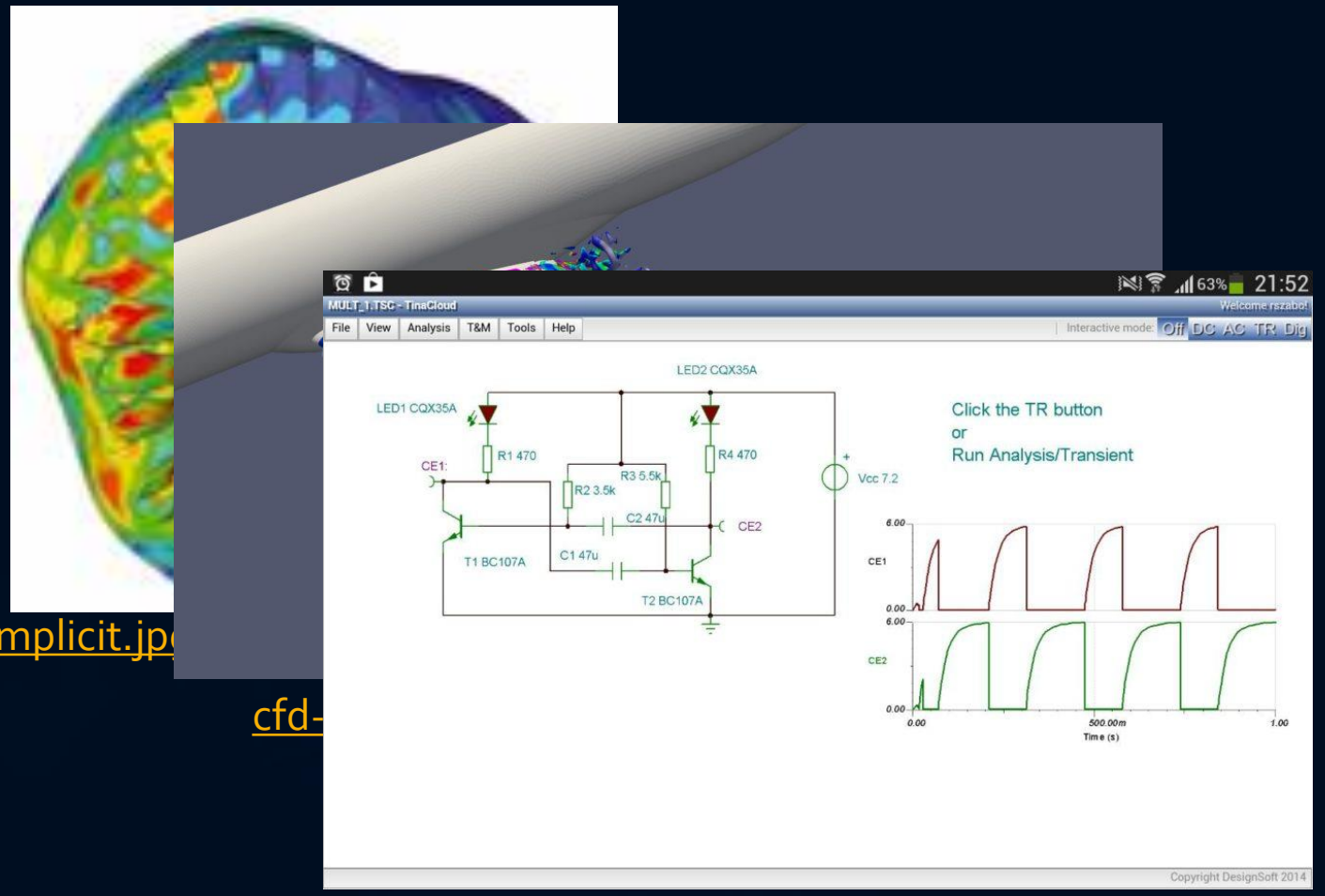
BPMN is an

analysis agnostic

system modeling language

Integrated Analyses

- FEA
- CFD
- Circuit simulation
- Modelica

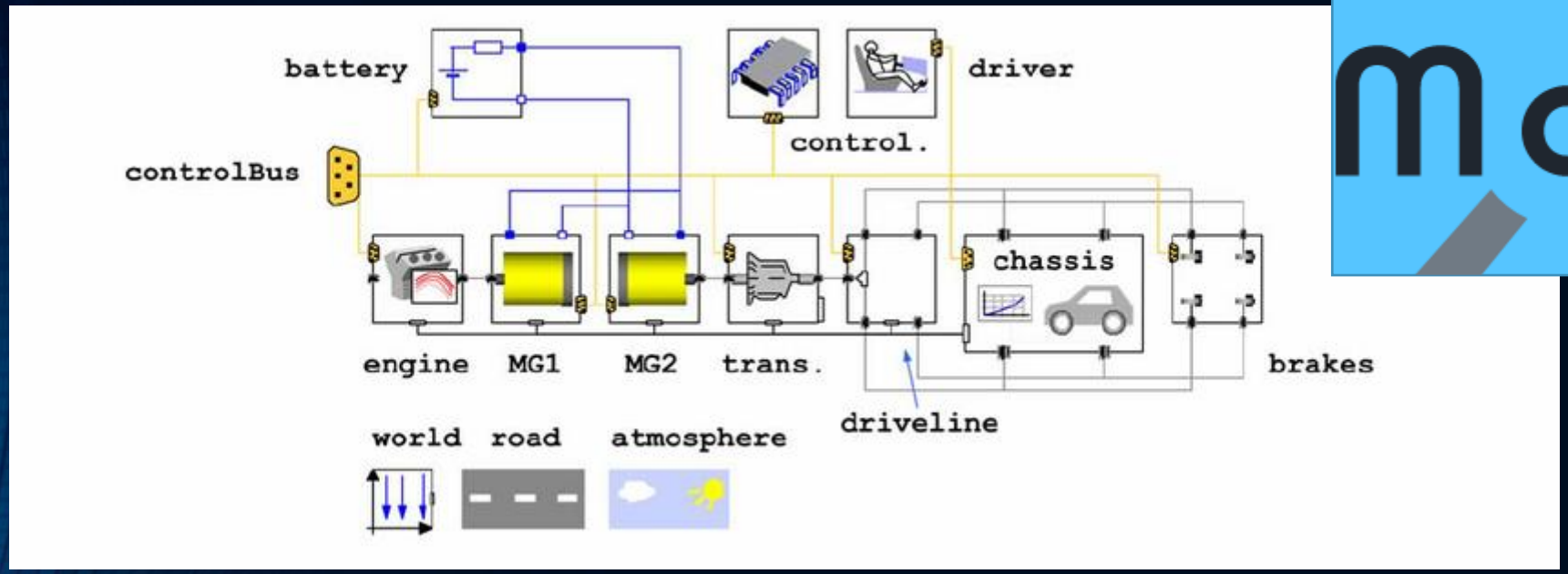


implicit.jp

cfd-

TinaCloud-1024x640.jpg (1024×640) (techyv.com)

Modelica



The-Modelica-simulation-model-of-the-series-hybrid-electric-vehicle.png (850×309) (researchgate.net)

Modelica: DSL + analysis-agnostic system model + integrated analysis

Recap: Successful DT technologies are based on:

- Domain-specific language
  - Informs instance system specification
  - Supports design methodology
- Integrated (or federated) analysis agnostic computational system models specifying observable structure and behavior
- Analyses integrated with system specification models
- ***Implementable computational theory of control***

Digital Twins for DELS

DELS Analysis Models

Analysis Tools

$L = \lambda W$
Activity network
Flow network
Queuing network
Simulation

For nearly 70 years, we've been analyzing DELS by creating *ad hoc* analysis models. The pinnacle of analysis tools is discrete event simulation models.

Observer DT

DigitalEnterprise_MES-1536

Digital Twin System in

GE Healthcare

Humber's Quality Command Centre Helps Deliver Better, Faster, and Safer Patient Care: Year One in Review

Case Study: Humber River Hospital's Command Centre Year One In Review - Points of View from GE Healthcare Command Centers (gehccommandcenter.com)

DigitalEnterprise_MES-1536x864.jpg (1536×864) (projectbinder.eu)

Observations:

- State
  - Resource (instrumented assets)
  - Units of flow (jobs, work orders, patients, etc)
- State change
  - Sensors
  - Transactions

Observation Requirements:

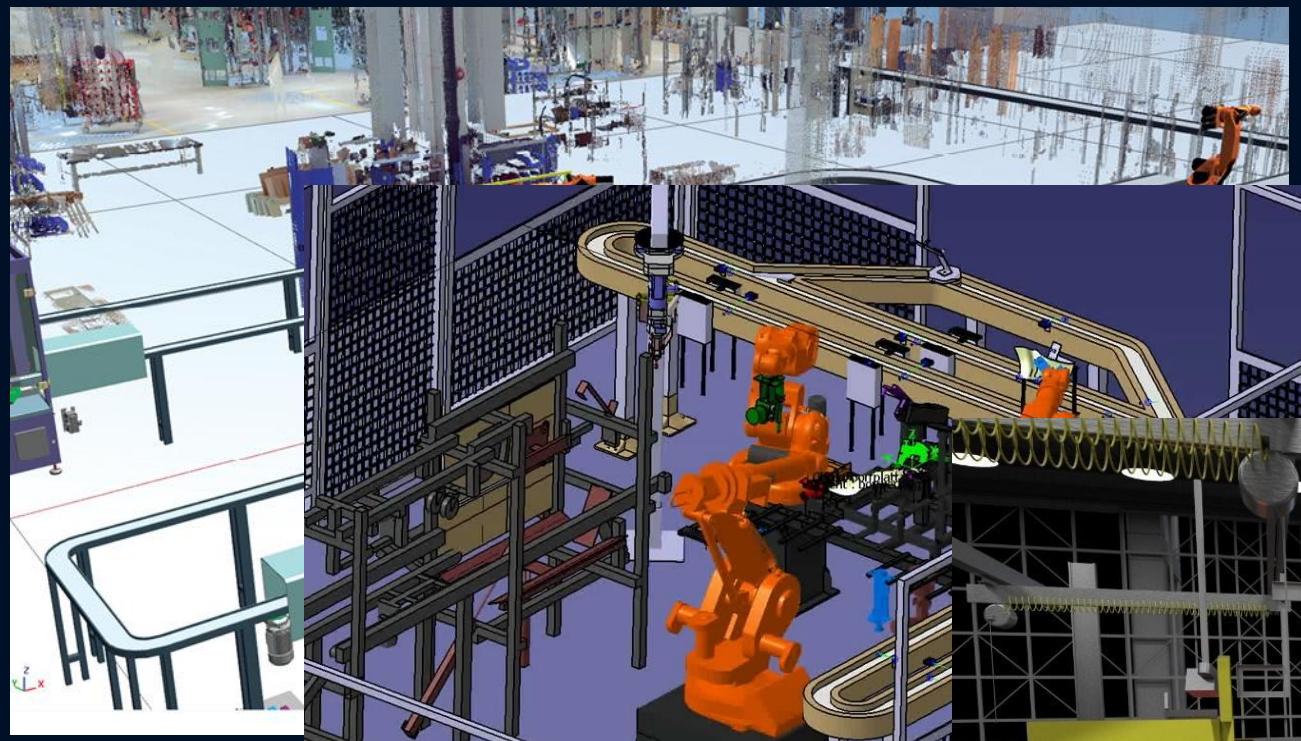
- Instrumentation
- Representation (visualization)

Sense-making

- Underlying explanatory ***system*** model

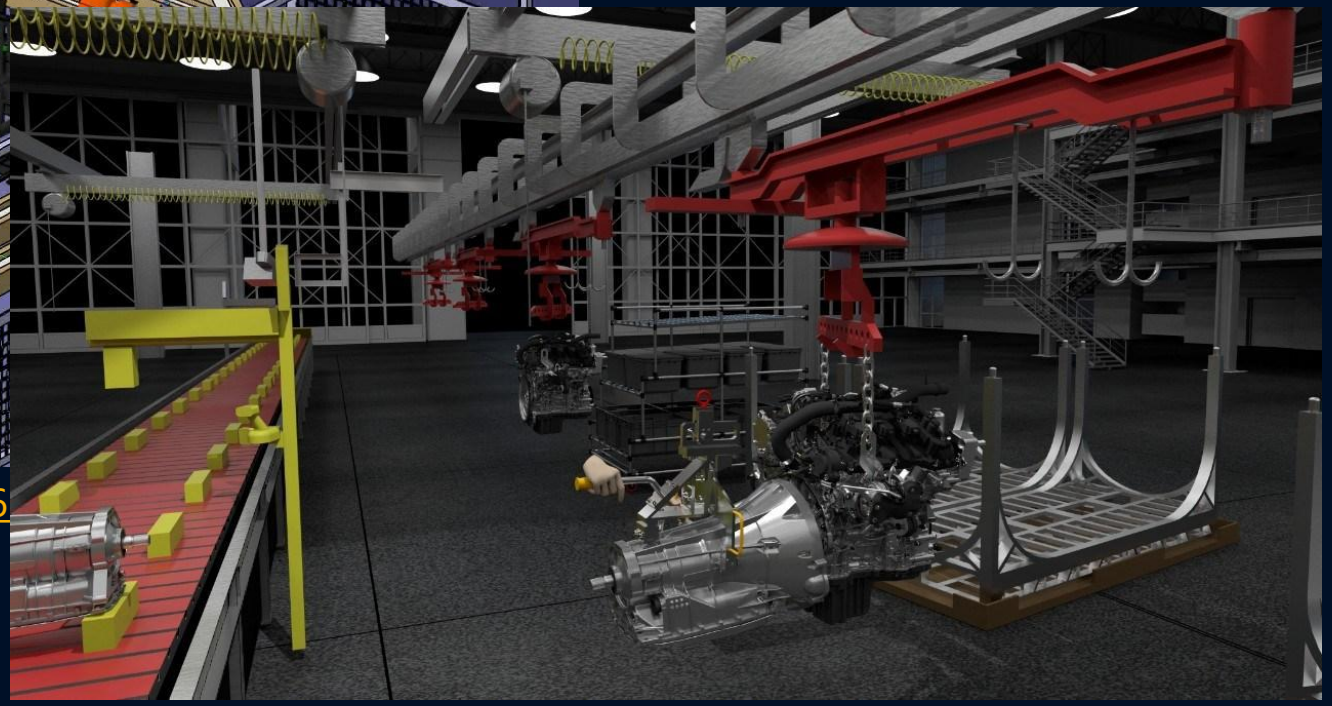
In what language is the underlying explanatory system model articulated?

Virtual DT



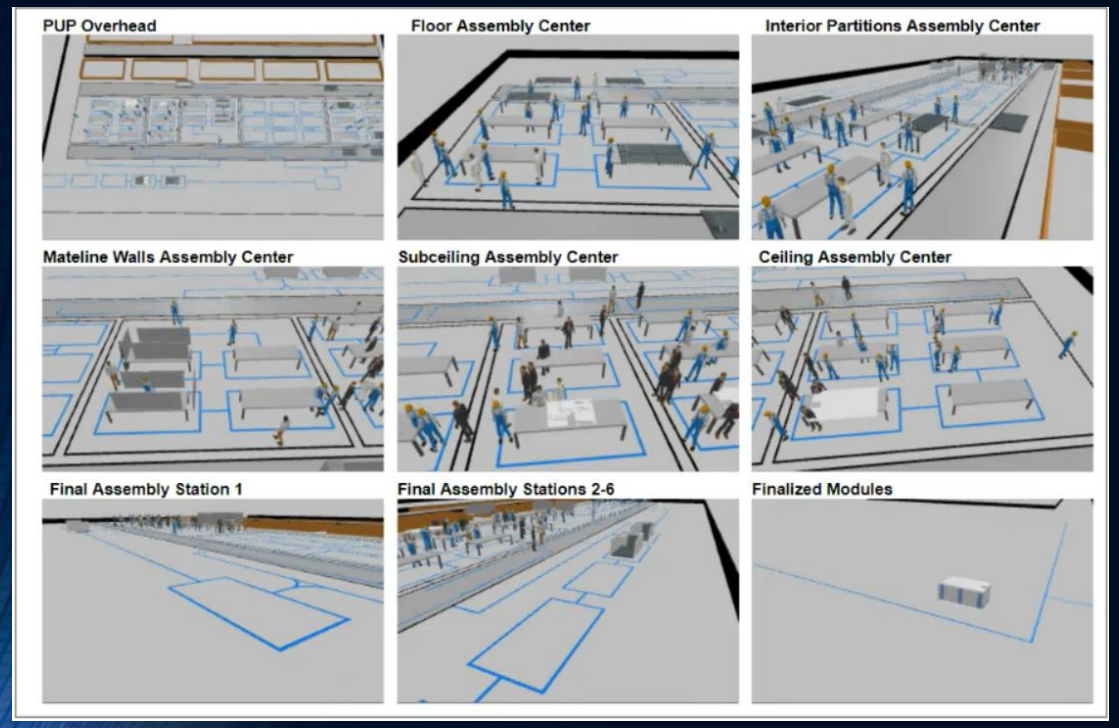
Siemens-Digit

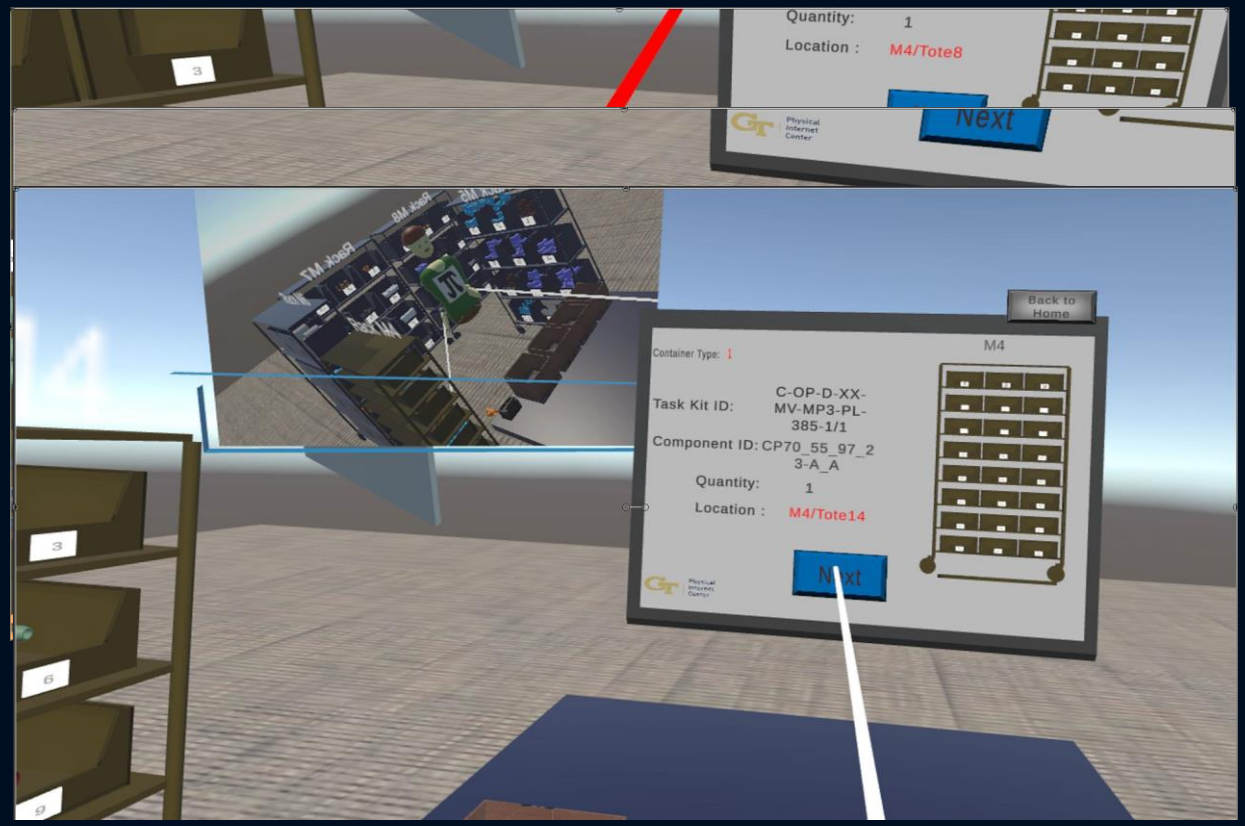
maxresdefault.jpg (1280×706



Ford Shows How Virtual Reality Will Change Our Lives (triplepundit.com)

Animation and Virtual Reality





legacy-ERP-money-pit.jpg (1200x1200) (plex.com)

What's missing?

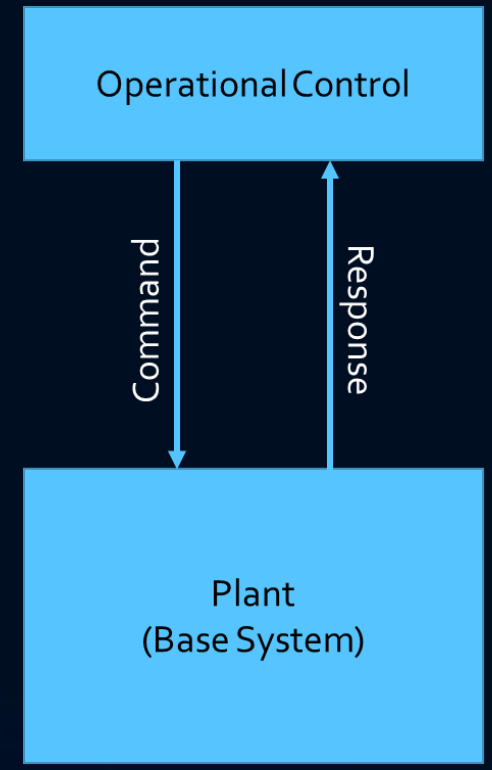
- DELS domain specific language (a la VHDL or BPMN)
- DELS theory of control (implementable)
- Analysis-agnostic DELS models (system specification)
- DELS Integrated analyses (performance, cost, quality, …)

What's missing?

- **DELS domain specific language (a la VHDL or BPMN)**
- DELS theory of control (implementable)
- Analysis-agnostic DELS models (system specification)
- DELS Integrated analyses (performance, cost, quality, …)

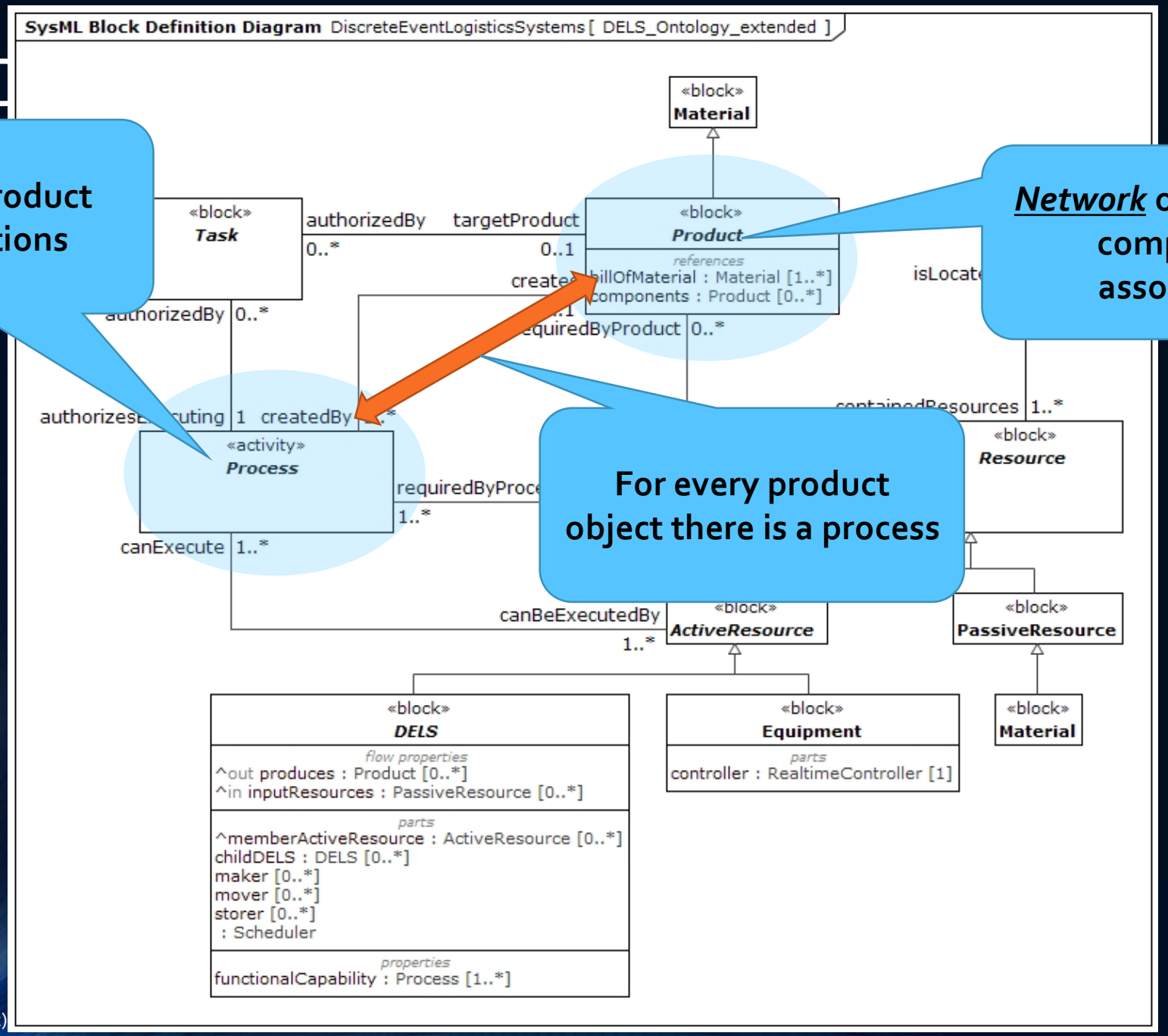
Toward a DSL for DELS

Basic conceptual framework



Directing Behavior

Structure + Behavior



Network of product transformations

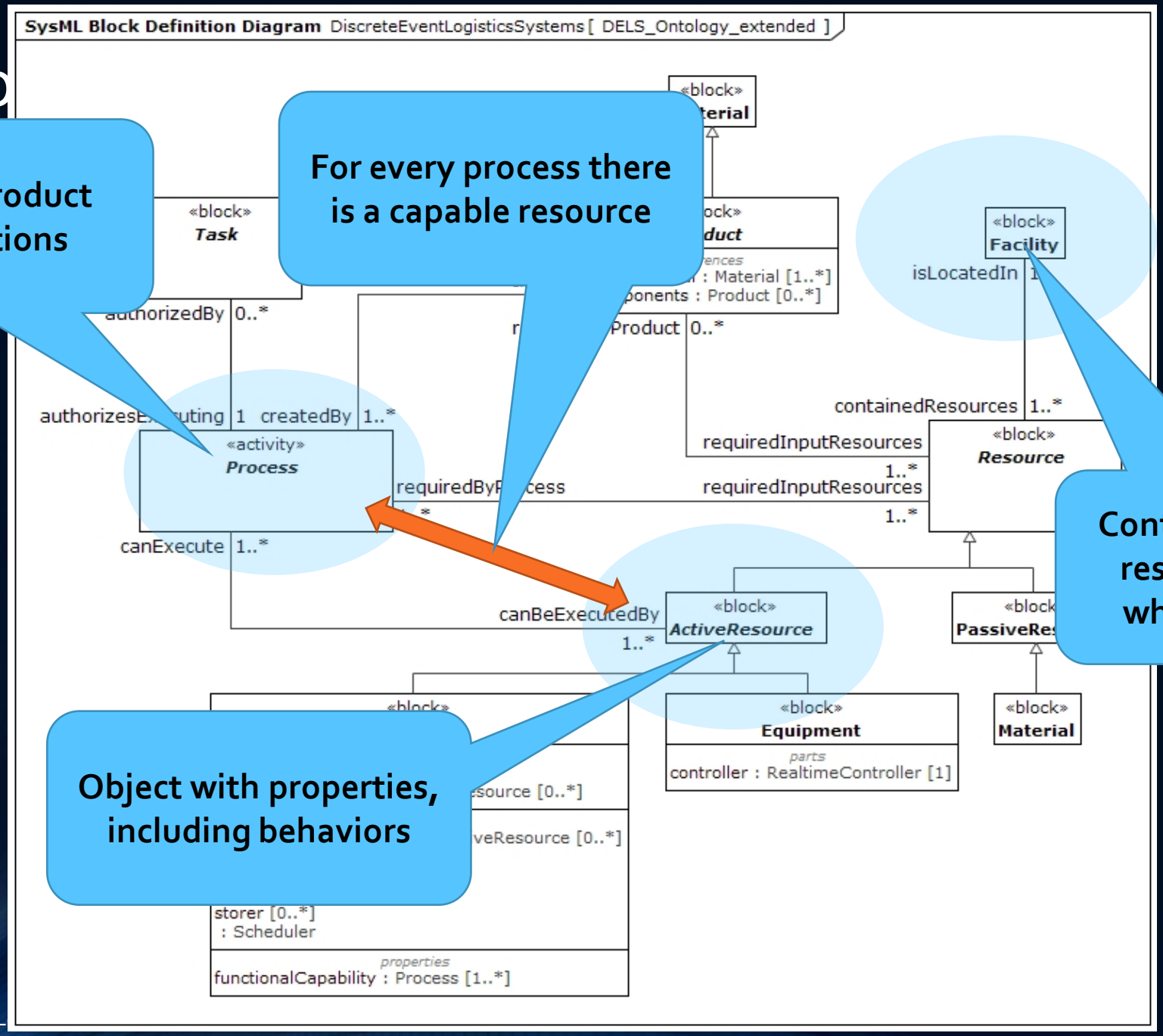
Network of objects and composition associations

For every product object there is a process

Do

Network **of product transformations**

For every process there is a capable resource



Contains a *network* of resources, through which objects flow

Object with properties, including behaviors

Formalizing a Reference Model

Theory of Discrete Event Logistics Systems (DELS) Specification

Published: June 12, 2020

Author(s)
Timothy A. Sprock, George Thiers, Leon F. McGinnis, **Conrad E. Bock**

Abstract
System models and model-based engineering methods have the promise of transforming the way that industrial engineers interact with production and logistics systems. Model-based methods play a role in improving communication between stakeholders, interoperability between systems, automated access to consistent analysis models, and multi-disciplinary design methods for complex systems. However, there remains a need for a foundation for modeling these kinds of systems -- a foundation that tailors methods and tools developed in other engineering domains to the unique concepts and semantics of production and logistics. This foundation is the topic of this report. This report documents a framework and model libraries for modeling discrete event logistics systems (DELS), an abstraction that covers manufacturing plants, material handling and transportation systems, warehouses, supply chains, etc. The DELS abstraction was created by identifing and modeling commonalities across the kinds of systems that industrial engineers typically encounter, and analysis models they use to analyze those system. It extends well-known product, process, and resource (PPR) ontologies to incorporate a library of operational control model components, and is connected to Commodity Flow Network (CFN), modeling networks, flow networks, and process networks. The relationship between DELS and CFN formally links system models to abstractions used to create analysis models, such as discrete event simulation.

Citation: NIST Interagency/Internal Report (NISTIR) - 8262

Report Number: 8262

NIST Pub Series: NIST Interagency/Internal Report (NISTIR)

Pub Type: NIST Pubs

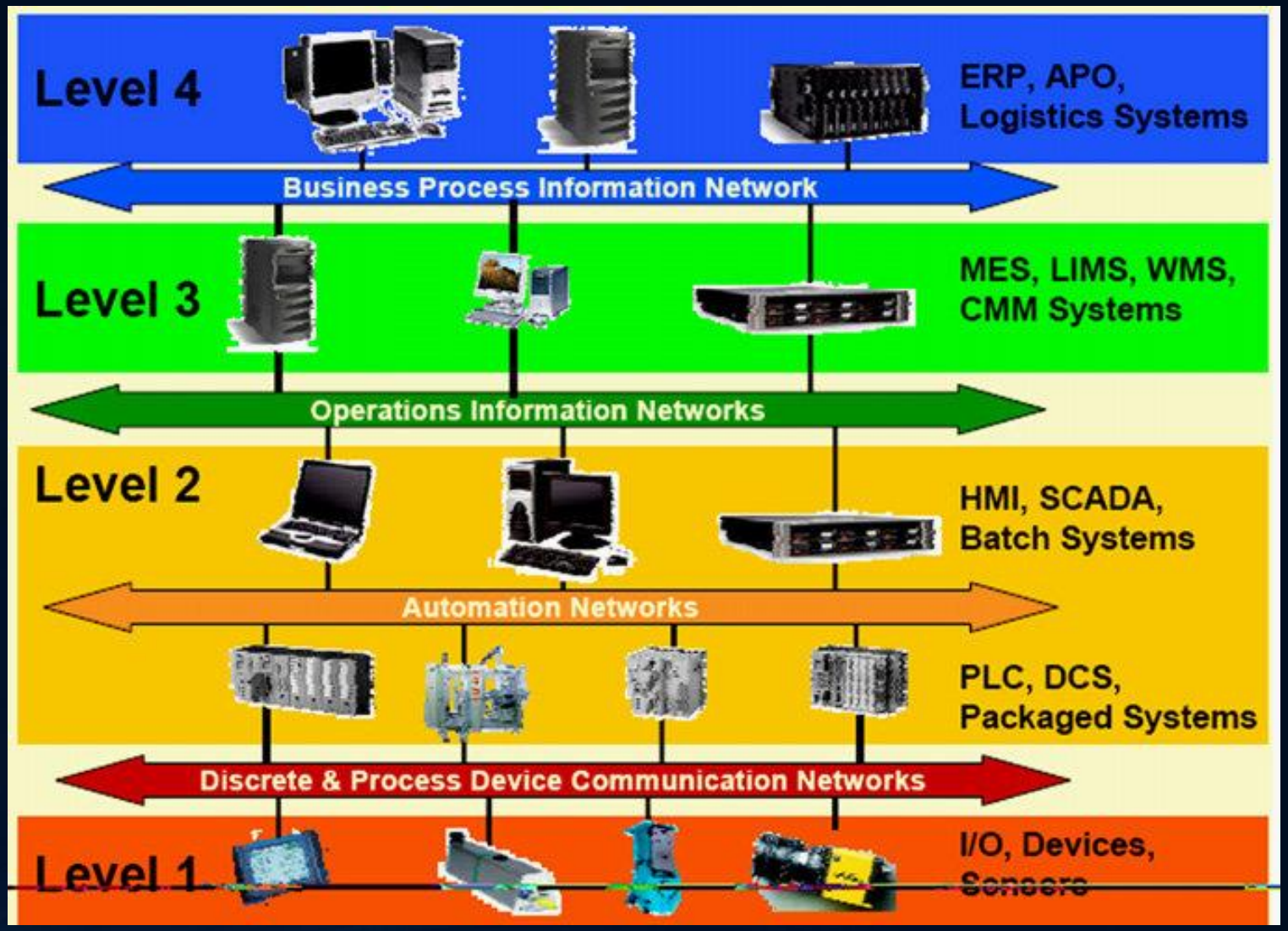
Download Paper
DOI Link

Theory of Discrete Event Logistics Systems (DELS) Specification | NIST

What's missing?

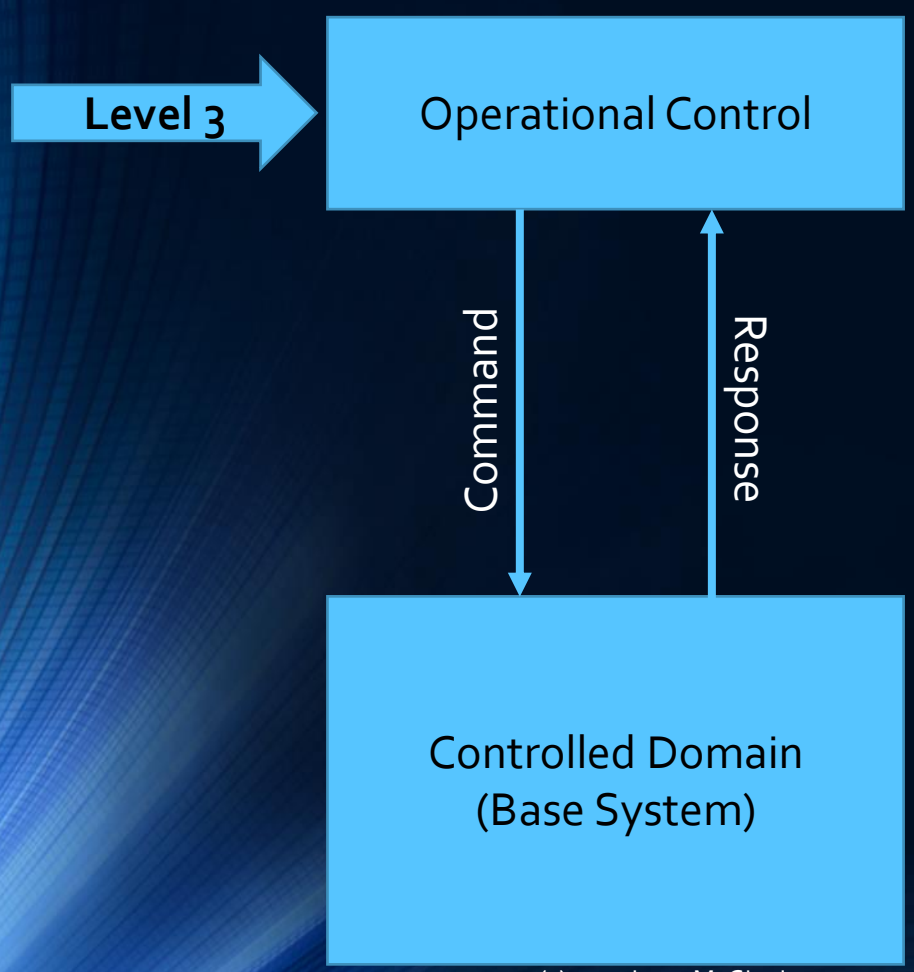
- DELS domain specific language (a la VHDL or BPMN)
- **DELS theory of control (implementable)**
- Analysis-agnostic DELS models (system specification)
- DELS Integrated analyses (performance, cost, quality, …)

Toward a Theory of DELS Operational Control



ISA95-control-hierarchy-14_W840.jpg (684×494) (researchgate.net)

L3 Controller Context



Modeling Focus

Directing Behavior

Structure + Behavior

Modeling Framework

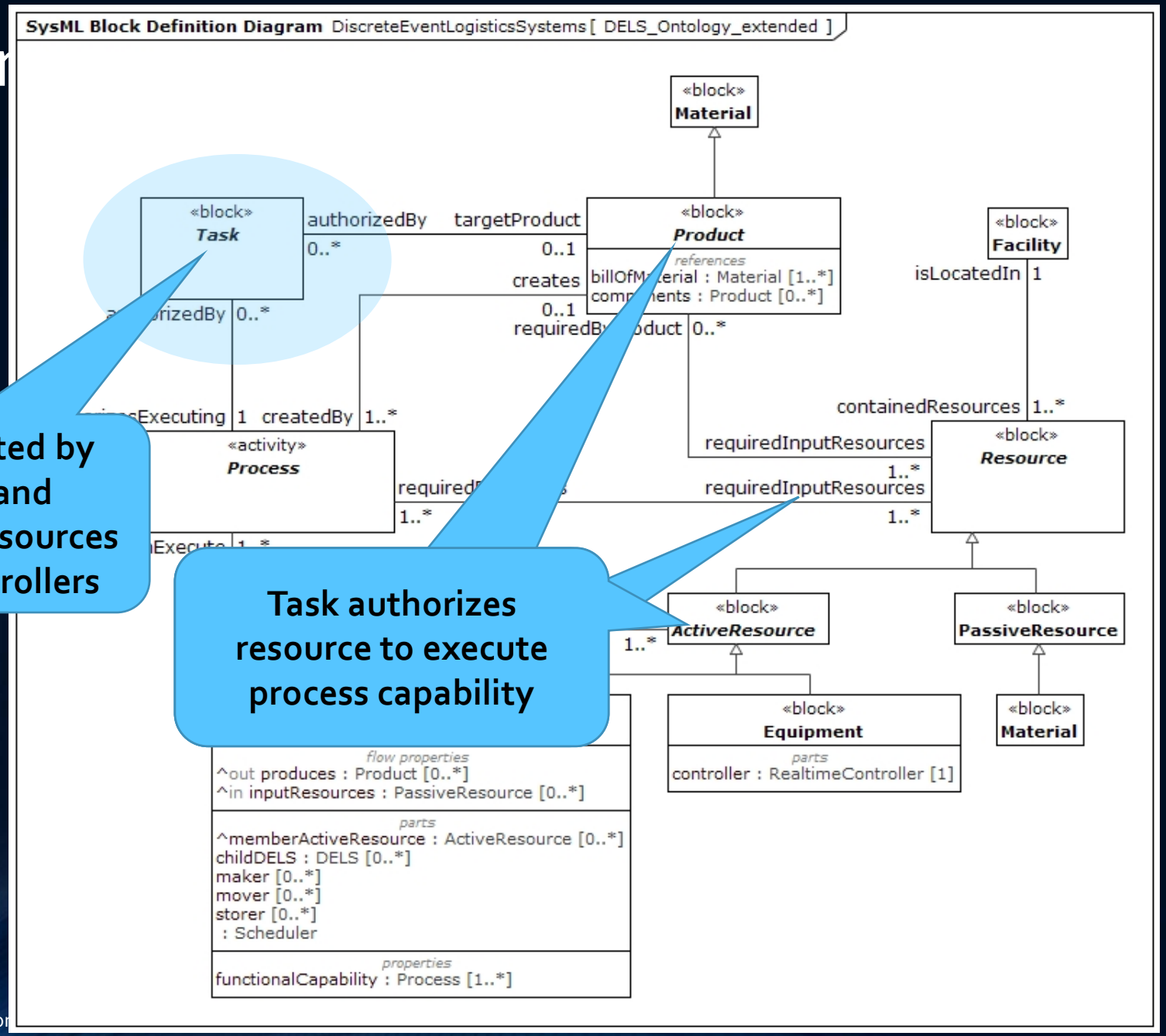
- Should a task be accepted?
- If so, when executed?
- With which resources?
- What happens next?
- Should a resource state change?

Product
Process
Resource
Facility

DELS Level 3 Operational Control Assumptions

- Decision execution is Event-driven
  - Received signals
  - Invocations of controller behavior
- Operational decisions are State-based
  - Decision-making has access to state of resources and tasks
- Operational control is imposed by invoking behaviors of active resources
  - In controller's domain
  - In other controllers' domains

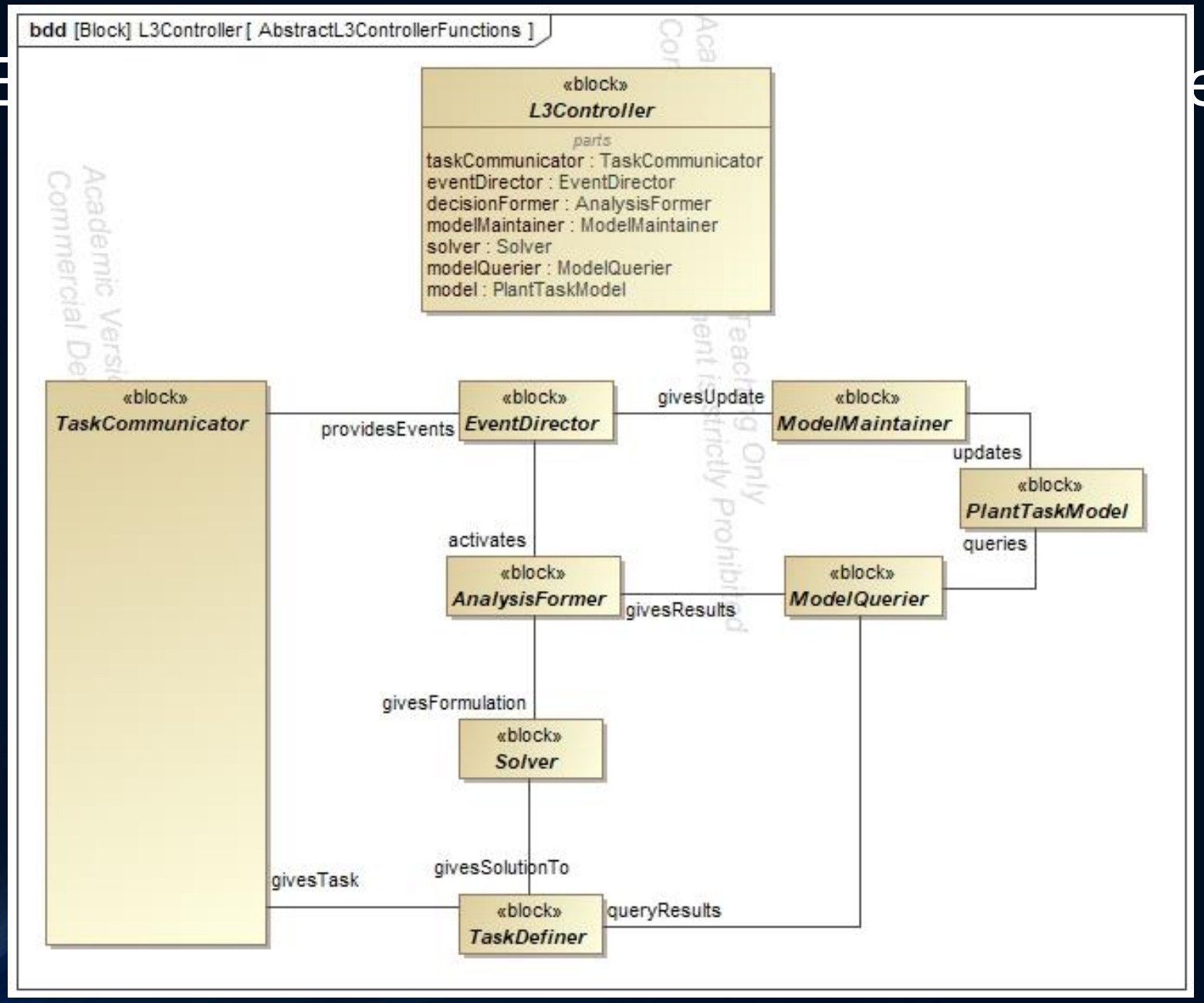
Dor

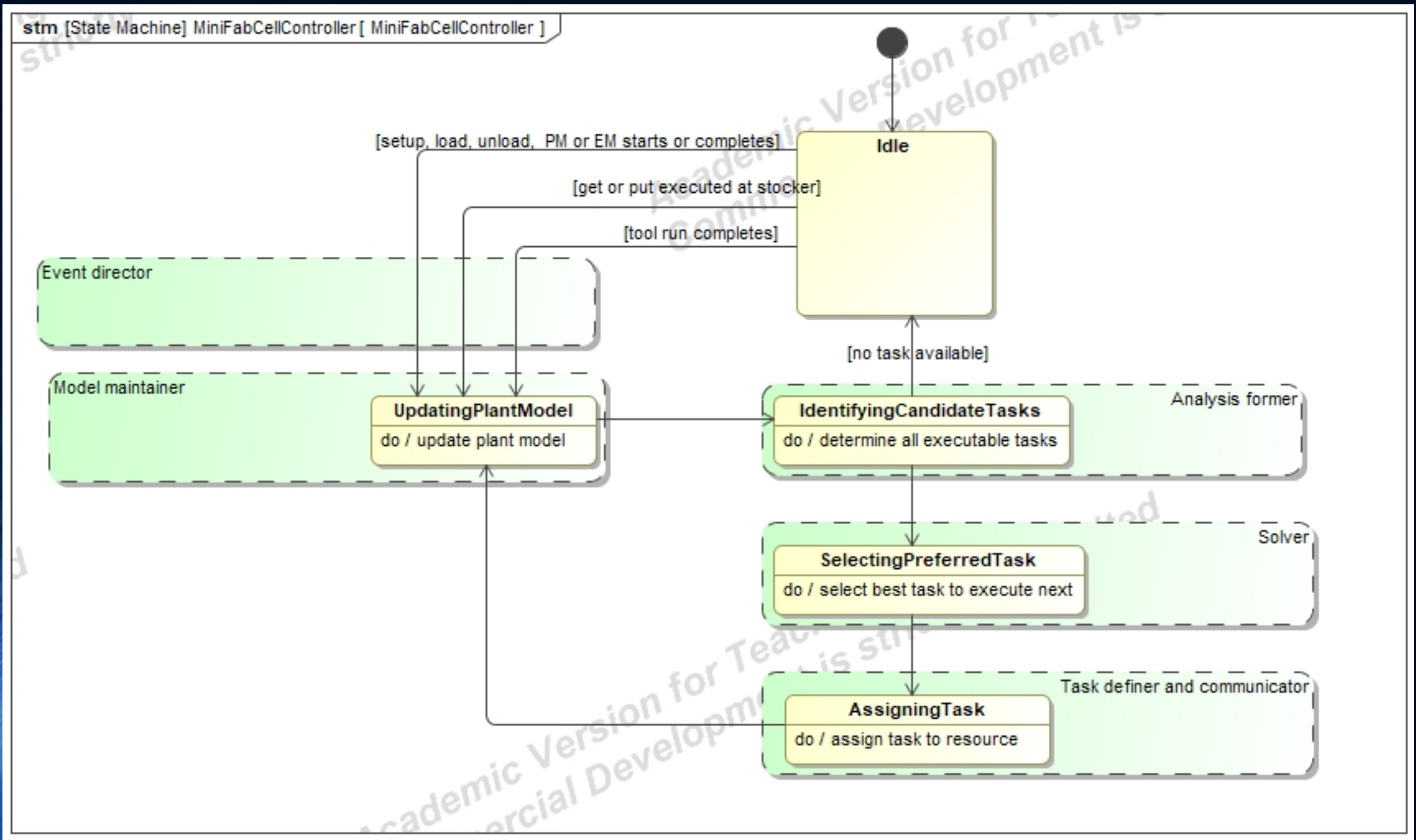


Tasks are created by controllers and consumed by resources and other controllers

Task authorizes resource to execute process capability

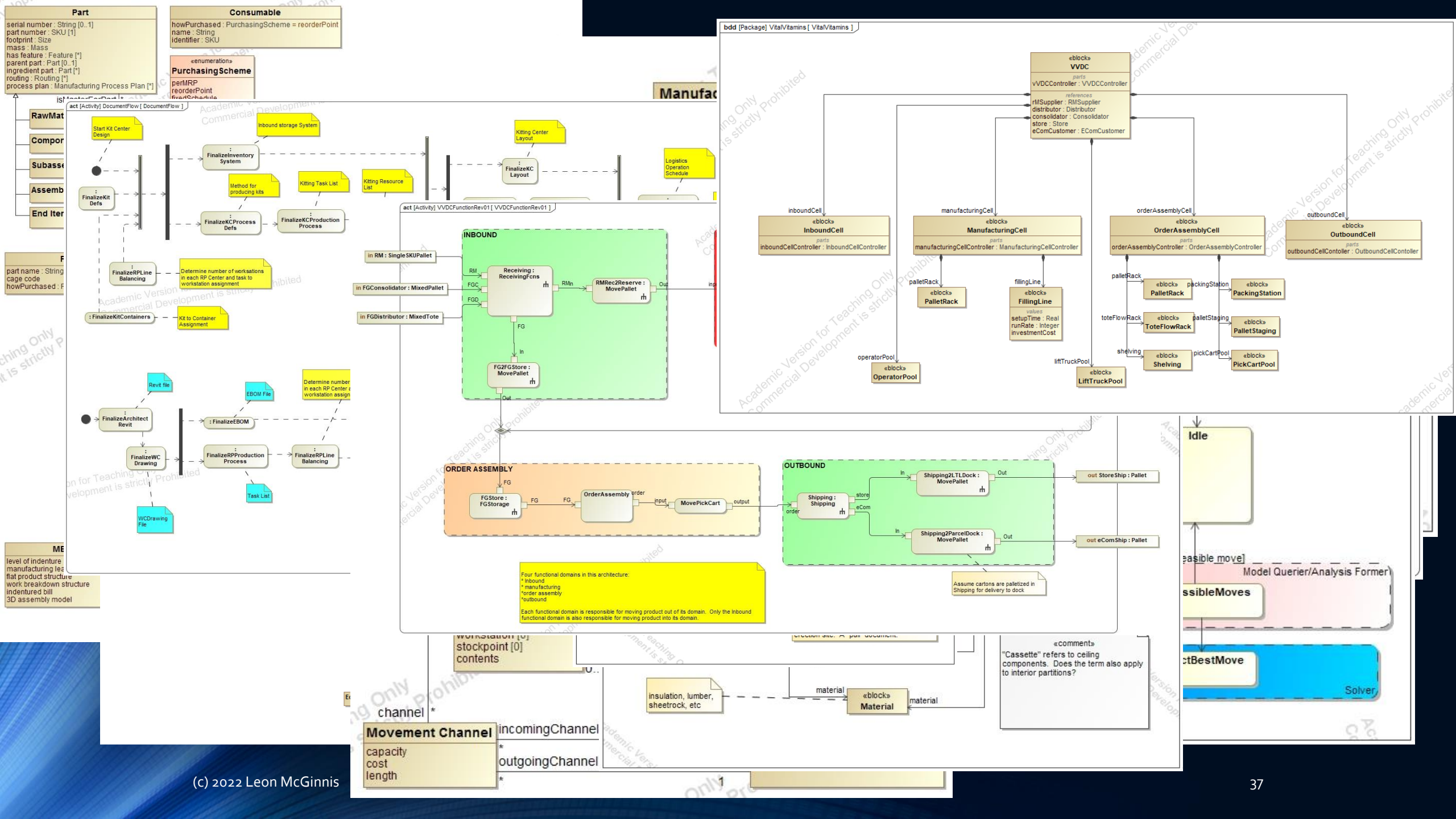
DE e




stm [State Machine] MiniFabCellController [MiniFabCellController]
[setup, load, unload, PM or EM starts or completes]
[get or put executed at stocker]
[tool run completes]
Idle
Event director
[no task available]
Model maintainer
UpdatingPlantModel
do / update plant model
Analysis former
IdentifyingCandidateTasks
do / determine all executable tasks
Solver
SelectingPreferredTask
do / select best task to execute next
Task definer and communicator
AssigningTask
do / assign task to resource

What's missing?

- DELS domain specific language (a la VHDL or BPMN)
- DELS theory of control (implementable)
- **Analysis-agnostic DELS models (system specification)**
- DELS Integrated analyses (performance, cost, quality, …)


Part
serial number : String [0..1]
part number : SKU [1]
footprint : Size
mass : Mass
has feature : Feature [*]
parent part : Part [0..1]
ingredient part : Part [*]
routing : Routing [*]
process plan : Manufacturing Process Plan [*]
Consumable
howPurchased : PurchasingScheme = reorderPoint
name : String
identifier : SKU
«enumeration»
PurchasingScheme
perMRP
reorderPoint
fixedSchedule
act [Activity] DocumentFlow [DocumentFlow]
Start Kit Center Design
Inbound storage System
FinalizeInventory System
Kitting Center Layout
FinalizeKC Layout
Logistics Operation Schedule
FinalizeKit Defs
Method for producing kits
Kitting Task List
Kitting Resource List
FinalizeKCProcess Defs
FinalizeKCProduction Process
FinalizeRPLine Balancing
Determine number of worksations in each RP Center and task to workstation assignment
: FinalizeKitContainers
Kit to Container Assignment
Revit file
EBOM File
FinalizeArchitect Revit
: FinalizeEBOM
FinalizeWC Drawing
FinalizeRPProduction Process
FinalizeRPLine Balancing
Task List
WCDrawing File
act [Activity] VVDCFunctionRev01 [VVDCFunctionRev01]
INBOUND
in RM : SingleSKUPallet
in FGConsolidator : MixedPallet
in FGDistributor : MixedTote
Receiving : ReceivingFcns
RMRec2Reserve : MovePallet
FG2FGStore : MovePallet
ORDER ASSEMBLY
FGStore : FGStorage
OrderAssembly
MovePickCart
OUTBOUND
Shipping : Shipping
Shipping2LTLDock : MovePallet
Shipping2ParcelDock : MovePallet
out StoreShip : Pallet
out eComShip : Pallet
Four functional domains in this architecture:
* inbound
* manufacturing
*order assembly
*outbound
Each functional domain is responsible for moving product out of its domain. Only the inbound functional domain is also responsible for moving product into its domain.
Assume cartons are palletized in Shipping for delivery to dock
bdd [Package] VitalVitamins [VitalVitamins]
«block»
VVDC
vVDCController : VVDCController
rMSupplier : RMSupplier
distributor : Distributor
consolidator : Consolidator
store : Store
eComCustomer : EComCustomer
inboundCell
«block»
InboundCell
inboundCellController : InboundCellController
manufacturingCell
«block»
ManufacturingCell
manufacturingCellController : ManufacturingCellController
orderAssemblyCell
«block»
OrderAssemblyCell
orderAssemblyController : OrderAssemblyController
outboundCell
«block»
OutboundCell
outboundCellContoller : OutboundCellContoller
palletRack
«block»
PalletRack
fillingLine
«block»
FillingLine
setupTime : Real
runRate : Integer
investmentCost
operatorPool
«block»
OperatorPool
liftTruckPool
«block»
LiftTruckPool
packingStation
«block»
PackingStation
toteFlowRack
«block»
ToteFlowRack
palletStaging
«block»
PalletStaging
shelving
«block»
Shelving
pickCartPool
«block»
PickCartPool
Idle
Model Querier/Analysis Former
ssibleMoves
ctBestMove
Solver
stockpoint [0]
contents
insulation, lumber, sheetrock, etc
material
«block»
Material
«comment»
"Cassette" refers to ceiling components. Does the term also apply to interior partitions?
channel
Movement Channel
capacity
cost
length
incomingChannel
outgoingChannel

At some point, we must have agreed-upon standards for defining objects and their behavior in the DELS domain, in the same way the ME community has developed the standards reflected in Modelica.

What is the appropriate level of detail?

What's missing?

- DELS domain specific language (a la VHDL or BPMN)
- DELS theory of control (implementable)
- Analysis-agnostic DELS models (system specification)
- **DELS Integrated analyses (performance, cost, quality, …)**

Experience so far:

- Analysis-agnostic system model in SysML is an excellent requirements document for simulation model development, at least for some simulation tools
  - Robotic logistics hub
  - Modular housing manufacturing plant
- It's possible, but fairly *ad hoc* to automate some analysis model generation from the SysML models
  - Mandrel lifecycle analysis for composite manufacturing

Wide-open opportunity

- Define canonical analysis models exploiting the network structure of DELS system models.

Canonical Models in OR

- An optimization problem has the canonical statement:

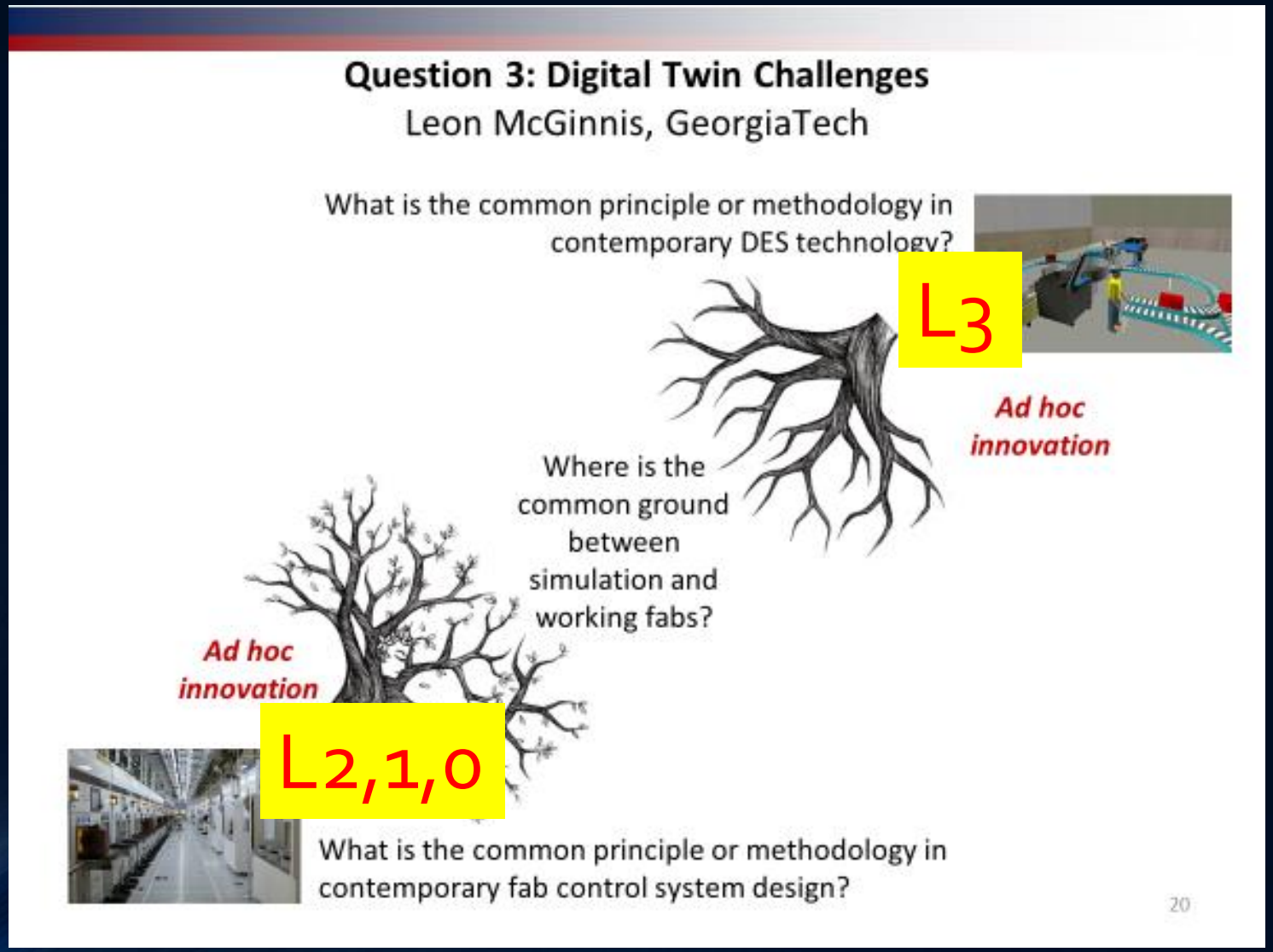
$$\begin{aligned} &\text{Min } cx \\ &\text{s.t. } Ax = b \\ &\qquad x \geq o \end{aligned}$$

The consequence is that we can have a "solver-independent" formulation of the analysis problem. The ontology and semantics are very straightforward.

- For discrete event simulation the canonical statement is:

The consequence is that the formulation of the analysis problem is always "solver-dependent". There currently is no generally accepted ontology and semantics for discrete event simulation (as generally practiced in IE domains).

Final thought



Four take-aways

- See digital twin idea in a historical context
  - Observer vs virtual prototype
- See fundamental challenges of DELS digital twins in that context
  - DSL, theory of control, analysis agnostic system models, analysis
- See potential approaches for resolving those challenges
  - DELS ontology, DELS L3 control theory, analysis integration
- See a major roadblock to ultimate success
  - Disconnect between L3 and the real-time systems at L2,1,0