2017 annual INCOSE international workshop
Los Angeles, CA, USA
January 28 - 31, 2017

System and Analysis Integration for Production & Logistics Systems

Conrad Bock[a], Leon McGinnis[b], & Timothy Sprock[a]
[a] National Institute of Standards and Technology, [b] Georgia Tech

Outline

- Digital Thread
- What are the fundamental challenges?
- Why & What are DELS
  - Commonalities First, Specifics Later
- Why is this interesting to the MBSE Initiative
- What do we want?

Digital Thread

- Digital Thread: platform for information to integrate product design, production and logistics systems design, and later stages of product lifecycle (sustainment)

- Design for Manufacturing: product/production design integration

- Production System Design Methodology: Processes, decision-making support, and analysis tools
  - Without a reference model you can't do it right today in a non ad-hoc way. Even with a reference model, you can't do it throughout the product's lifecycle since all of the analysis models have to be built by hand.

The SE “Vee” for both product & process

System Development	System Process Development
Con Ops	Global supply chain concept
Requirements/ Architecture	Technical capabilities and capacities, SC architecture
Detailed Design	Sourcing plan, facility design, planning/control concepts
Implement	Virtualize, test concepts, program roll-out
Integrate, Test, Verify	Global SC simulation, contingency analyses, standards, …
System V&V	Deployment
Operations & Maintenance	Operations

Computational support

CAD, FEA, CFD, PDM/PLM, REQUIREMENTS, SysML, and many more; increasing levels of integration and interoperability

Use models to specify, analyze, integrate, simulate, verify, validate—virtually, across disciplines

Excel, Visio, some CAD, optimization, simulation; not integrated, not interoperable

Use documents to specify and communicate, independent *ad hoc* models to support decision making

…velopment	System Process Development
… Ops	Global supply chain concept
…irements/ A… tecture	Technical capabilities and capacities
Deta… ed Design	SC architecture, sourcing plan, facility design, planning/control concepts
Implement	Virtualize, program roll-out
Integrate, Test, Verify	Global SC simulation, contingency analyses
System V&V	Deployment
Operations & Maintenance	Operations

Fundamental Challenges

- (Lack of) Common semantics & syntax for specifying production systems *(reference model)*
  - Difficulty of integration in PDM/PLM systems

- Time and expense of hand-coding analysis models (imagine if every FEA/CFD required a simulation engineer to hand-code the model)
  - Very limited decision support to production system engineers

- (Lack of) An engineering design methodology for production systems
  - Very difficult to capture/re-use learnings from experience—lots of tacit rather than explicit knowledge

What are DELS?

Discrete event logistics systems (DELS) are a class of dynamic systems that are defined by the transformation of discrete flows through a network of interconnected subsystems.

- These systems share a common abstraction, i.e. *products* flowing through *processes* being executed by *resources* configured in a *facility* (PPRF).

Examples include:

- Supply chains
- Manufacturing systems
- Transportation
- Material handling systems
- Storage systems
- Humanitarian logistics
- Healthcare logistics
- Semiconductor manufacturing
- Reverse and Remanufacturing Logistics
- And many more …

- Fundamentally, these systems are very similar, and often DELS are actually composed of other DELS.
- This similarity (and integration) produces a common set of analysis approaches that are applicable across the many systems in the DELS domain.

Interest to MBSE Community

- Bring a different domain into the INCOSE community
  - In the design of logistics systems, we don't have good SE tools and practices
- Why can INCOSE have a big impact on this domain?
  - In addition to the SE best practices, MBSE has been transformative!
  - Explicit modeling and design methods
    - Consensus on how we talk about our artifacts and design them
  - Want to learn from MBSE community
- What are the things we need to do to have an impact:
  - Reference models, common design process, conforming and supporting analysis models and tools.
  - Build a community around a shared vision of DELS MBSE

It’s (long past) time to bring the power of (model based) systems engineering to production systems and global supply chains!

What does it take to do that?

Where are we in the journey?

Tuesday @ 8:10am in MBX/Ecosystems

leon.mcginnis@isye.gatech.edu
timothy.sprock@nist.gov
conrad.bock@nist.gov

2017 annual INCOSE international workshop
Los Angeles, CA, USA
January 28 - 31, 2017

MBSE for Discrete Event Logistics Systems (DELS)

Conrad Bock[a], Leon McGinnis[b], & Timothy Sprock[a]
[a] National Institute of Standards and Technology, [b] Georgia Tech

It’s (long past) time to bring the power of (model based) systems engineering to production systems and global supply chains!

What does it take to do that?

Where are we in the journey?

Outline

- What are DELS?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
  - What are the types of analysis models and problems we're interested in for DELS (SAI)?
- Where are we now?
  - What is contained in the DELS reference model?
  - System-Analysis Integration Use Case
- Where do we want to go?

What are DELS?

Discrete event logistics systems (DELS) are a class of dynamic systems that are defined by the transformation of discrete flows through a network of interconnected subsystems.

- ➢ These systems share a common abstraction, i.e. *products* flowing through *processes* being executed by *resources* configured in a *facility* (PPRF).

Examples include:

- Supply chains
- Manufacturing systems
- Transportation
- Material handling systems
- Storage systems
- Humanitarian logistics
- Healthcare logistics
- Semiconductor manufacturing
- Reverse and Remanufacturing Logistics
- And many more …

- ➢ Fundamentally, these systems are very similar, and often DELS are actually composed of other DELS.
- ➢ This similarity (and integration) produces a common set of analysis approaches that are applicable across the many systems in the DELS domain.

Fundamental Challenges

- (Lack of) Common semantics & syntax for specifying production systems *(reference model)*
  - Difficulty of integration in PDM/PLM systems

- Time and expense of hand-coding analysis models (imagine if every FEA/CFD required a simulation engineer to hand-code the model)
  - Very limited decision support to production system engineers

- (Lack of) An engineering design methodology for production systems
  - Very difficult to capture/re-use learnings from experience—lots of tacit rather than explicit knowledge

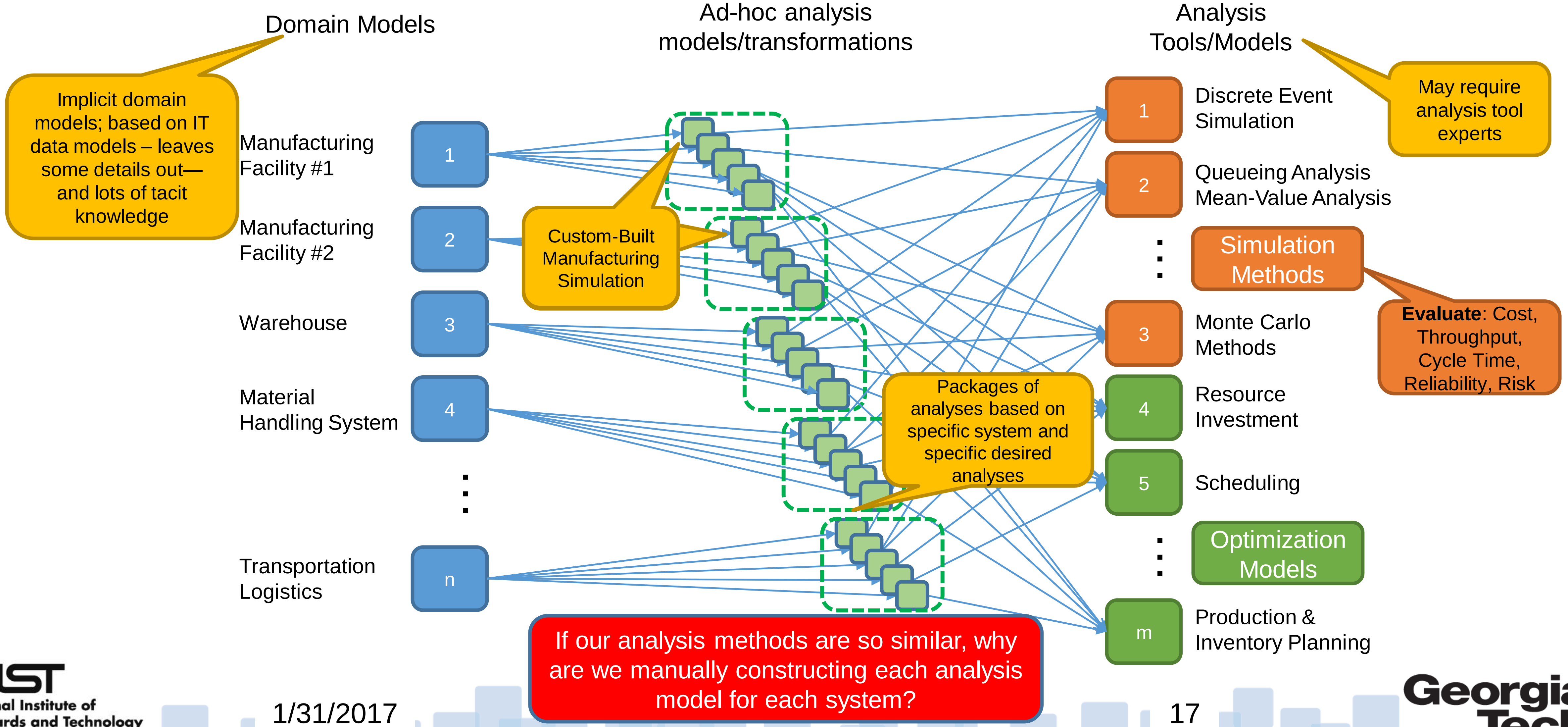
Outline

- What are DELS?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
    - What are the types of analysis models and problems we're interested in for DELS (SAI)?
- Where are we now?
    - What is contained in the DELS reference model?
    - System-Analysis Integration Use Case
- Where do we want to go?

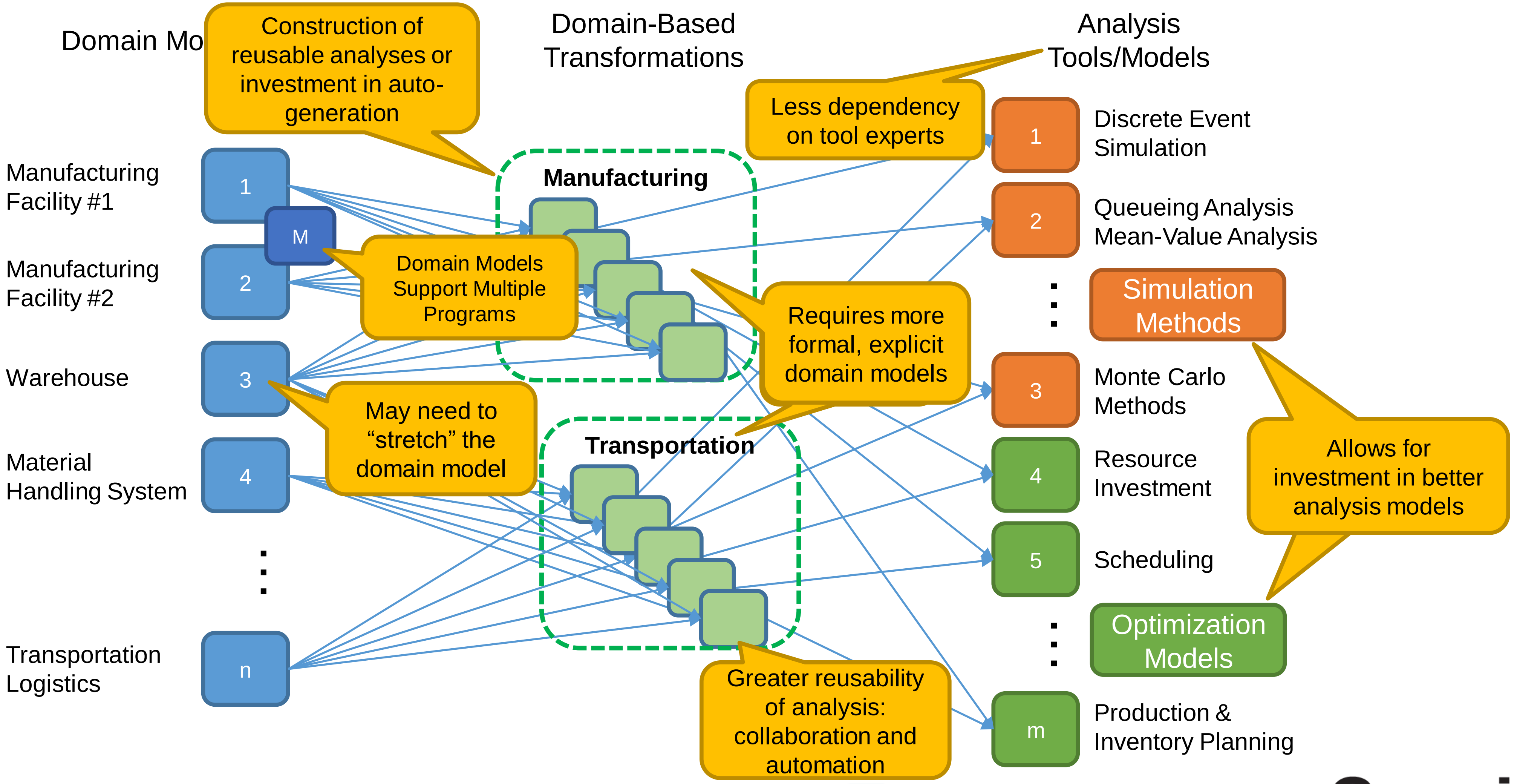
Need for Model-Based Methods

- Model-based methods: design, formal specification, and analysis
- Teaching, designing, automation
- Documentation & Organization of Knowledge
    - Existing Systems Models (industry)
        - We’ve had industrial partners tell us that the system model we built was the first time a particular system had been document, or most complete documentation of a particular system.
    - Existing Analysis Models (academia)
        - Why do we make particular assumptions or abstractions?
        - Why can’t one method be applied to a different but similar problem?
        - Taxonomy of extensions and mapping to applicable use cases
- SAI – Bridge between system and analysis models
    - Interoperability between different analysis models of the same system

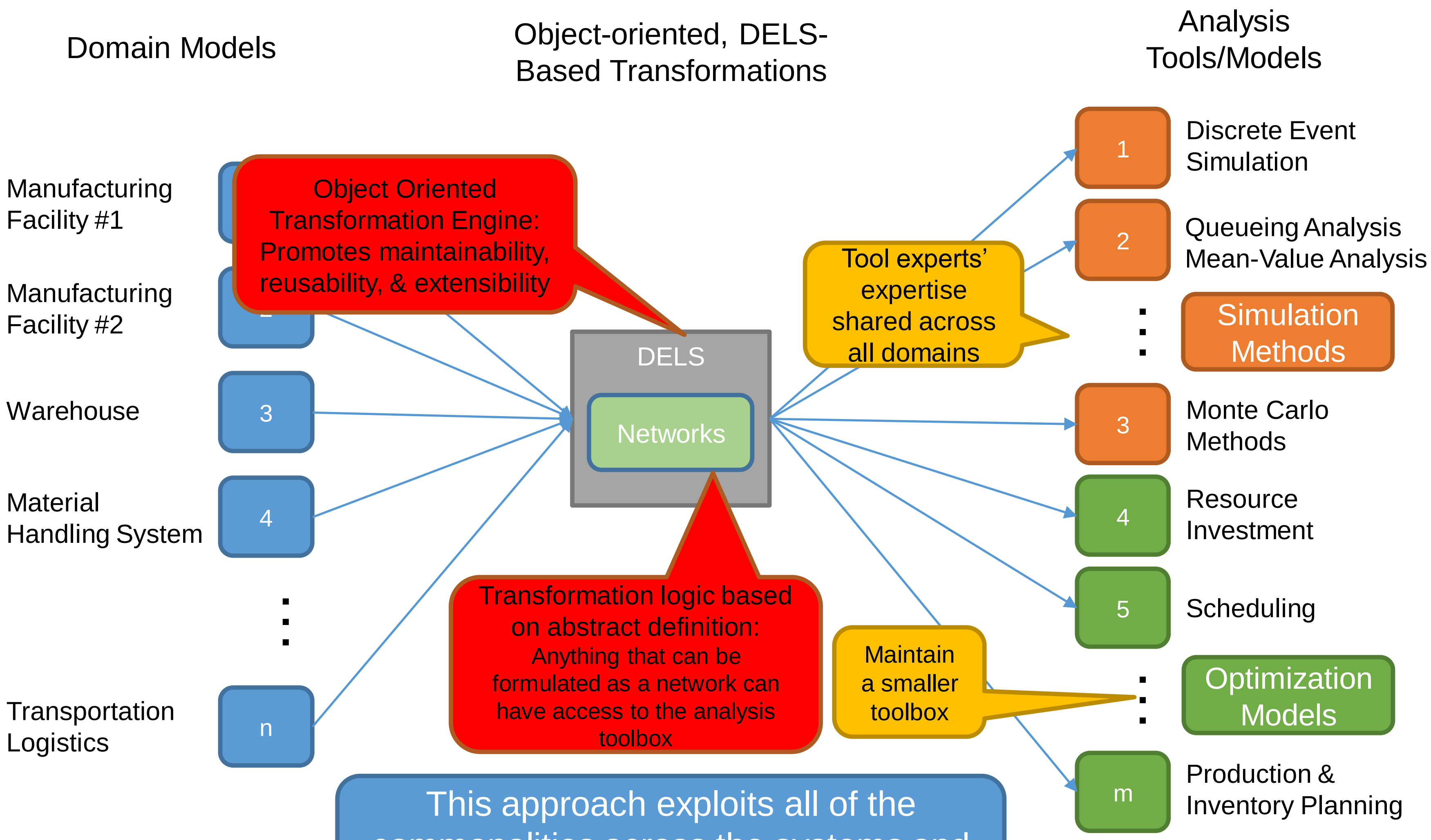
System Model to Analysis Model Transformation: Status Quo – Manual Ad-Hoc Analysis Generation



System Model to Analysis Model Transformation: M2M Methods Based on Domain Models

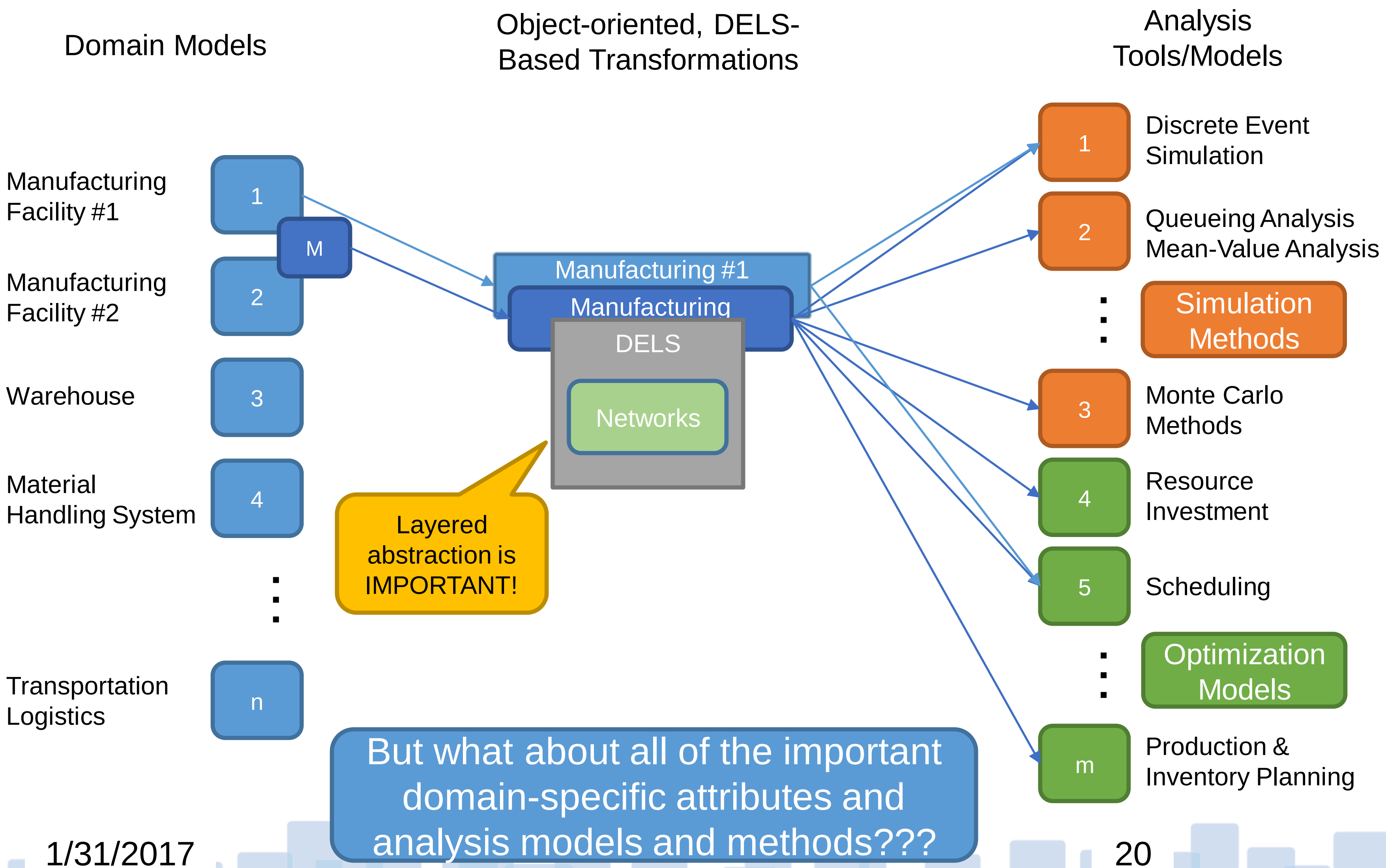


System Model to Analysis Model Transformation: M2M Methods Based on DELS Abstraction



This approach exploits all of the commonalities across the systems and analysis domains…

System Model to Analysis Model Transformation: *Extending* M2M Methods Based on DELS Abstraction



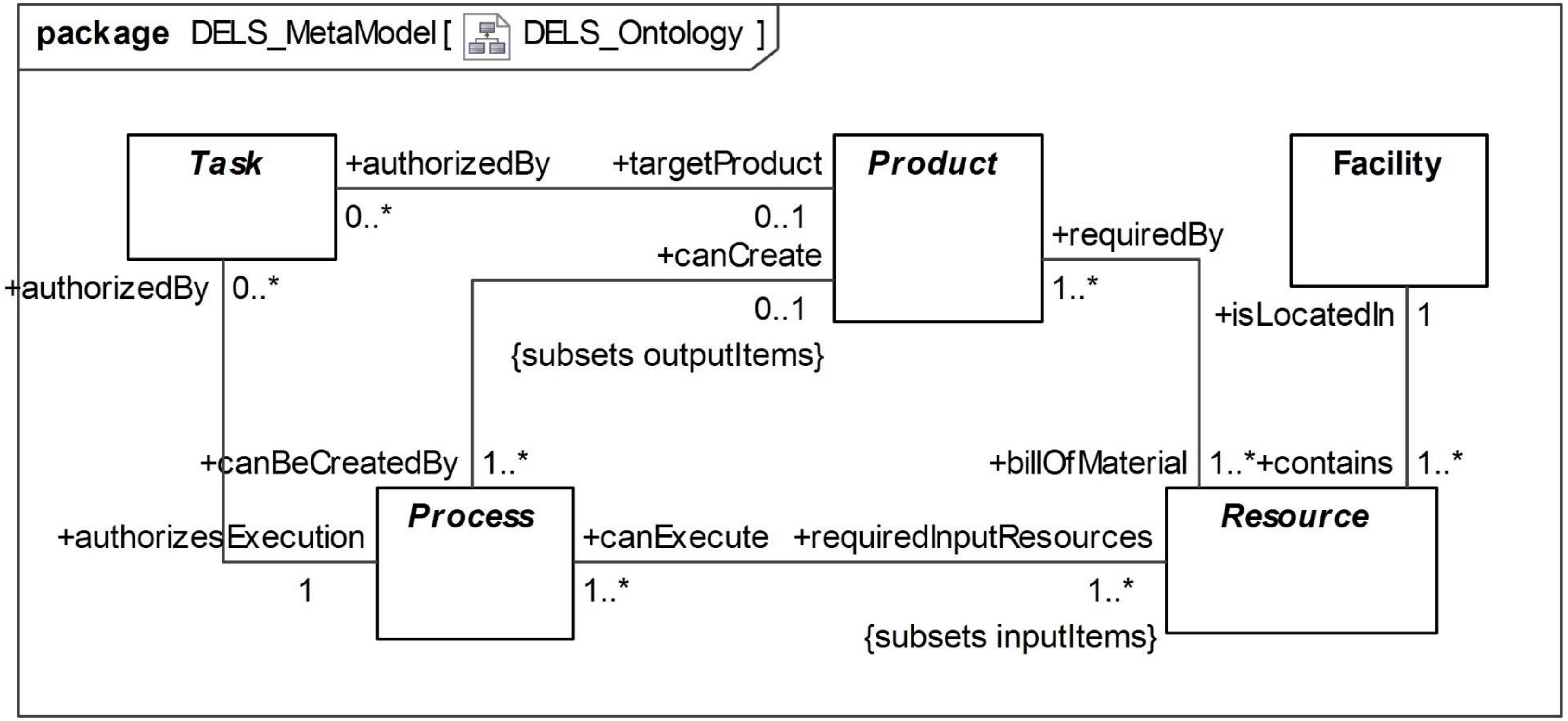
Outline

- What are DELS?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
  - What are the types of analysis models and problems we're interested in for DELS (SAI)?
- **Where are we now?**
  - What is contained in the DELS reference model?
  - System-Analysis Integration Use Case
- Where do we want to go?

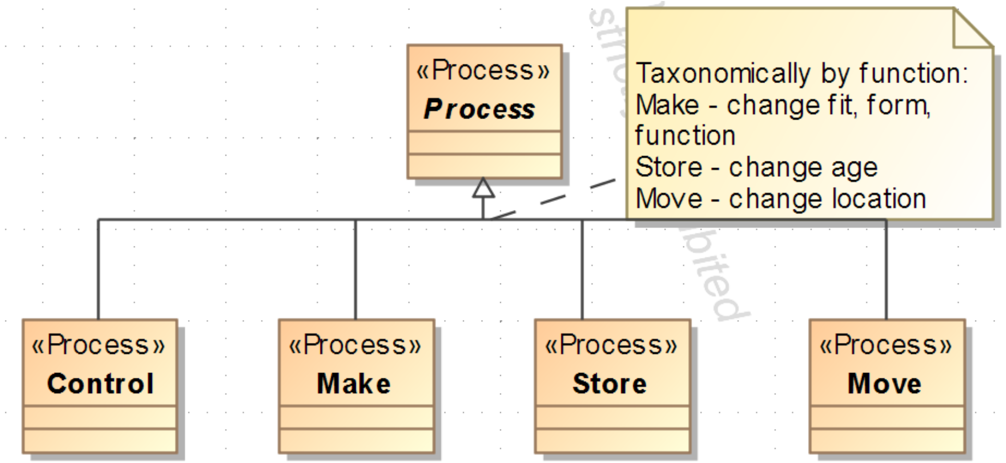
DELS Reference Model

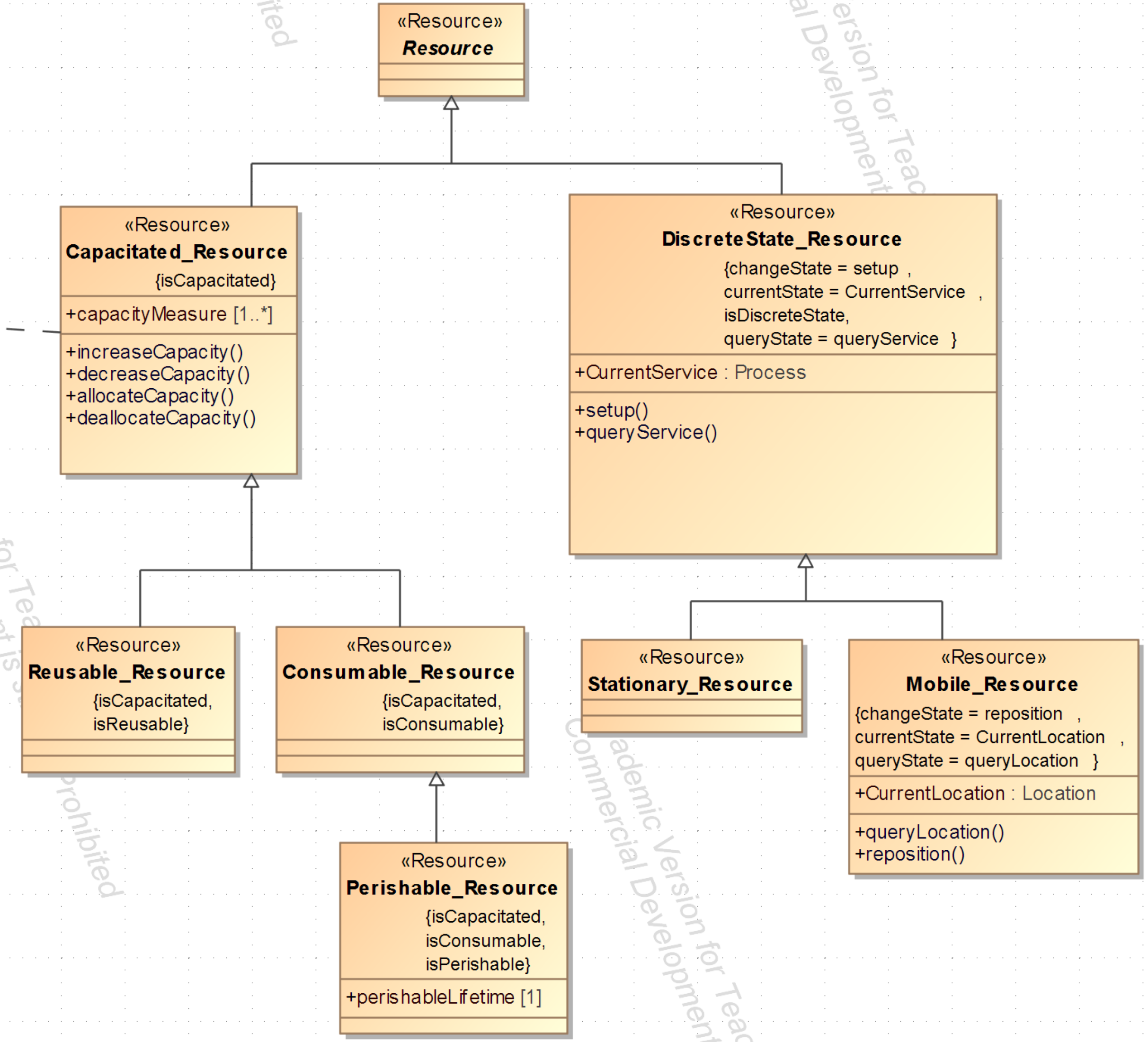
- Network Abstraction (Structural)
  - Abstraction: Networks, Flow Networks, Process Networks
- System Behavior (Plant)
  - Abstraction: Product, Process, Resource, Facility + Task
- Control
  - Admission, Sequencing, Resource Assignment, Routing, & Resource State
- Domain-specific Reference Models
  - Production (Make), Warehousing (Store), Transportation (Move)
  - Supply Chains, Healthcare Logistics, etc.

DELS Behavior – Product, Process, Resource, Facility

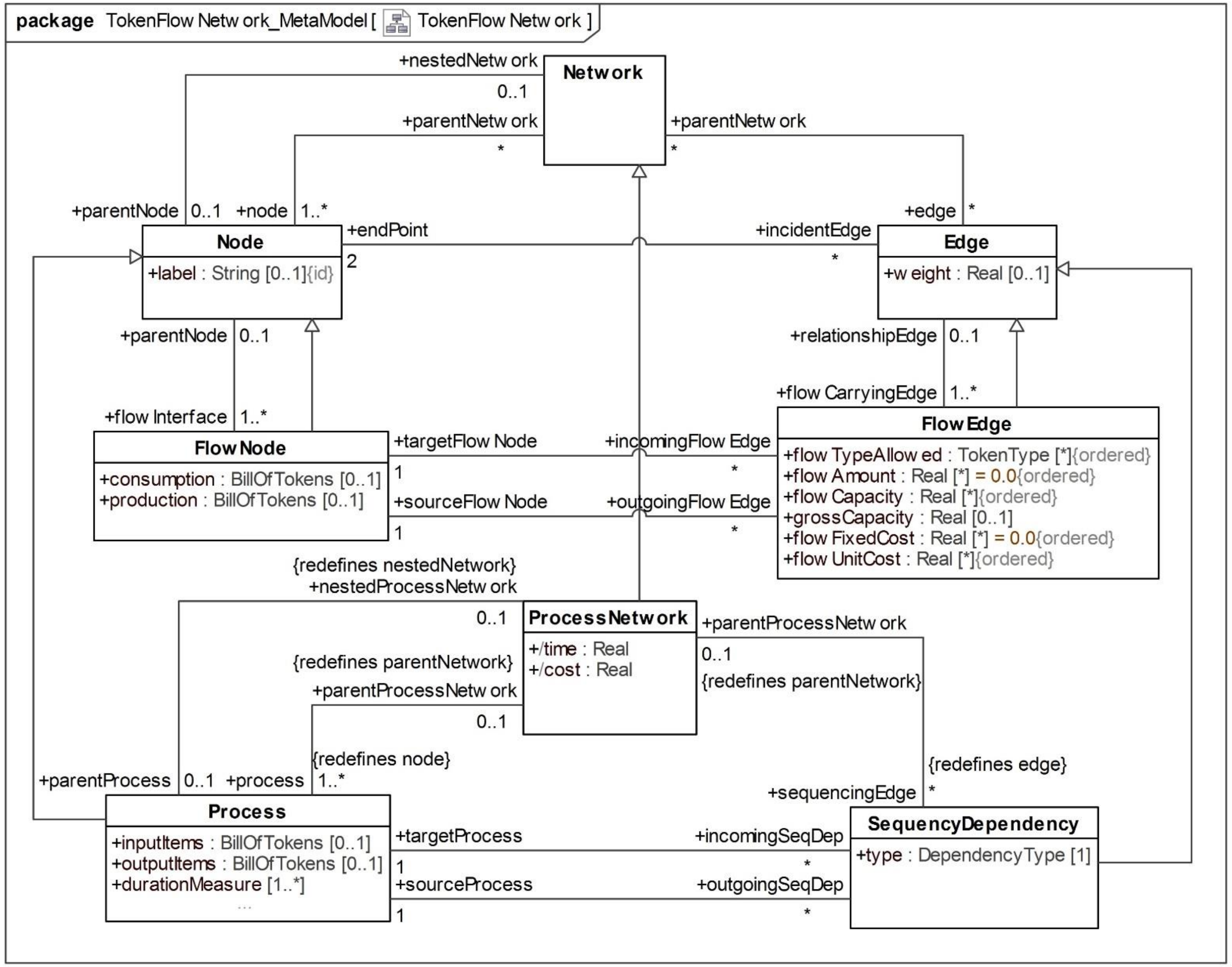


Taxonomies of DELS Behavior



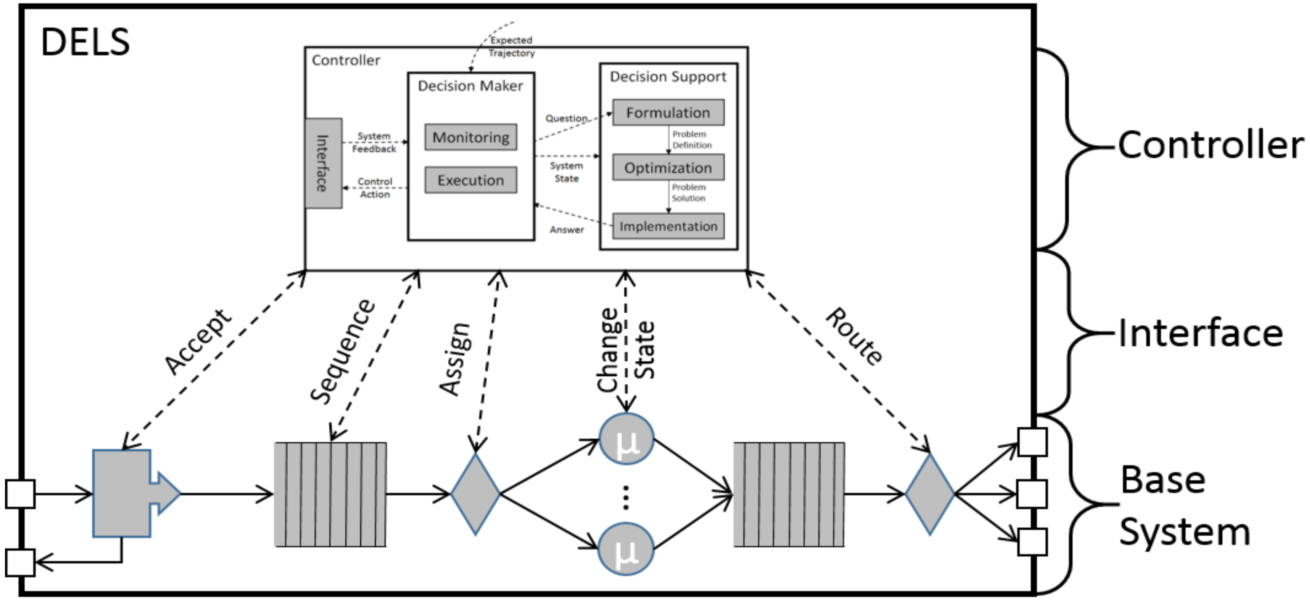


Network Abstraction



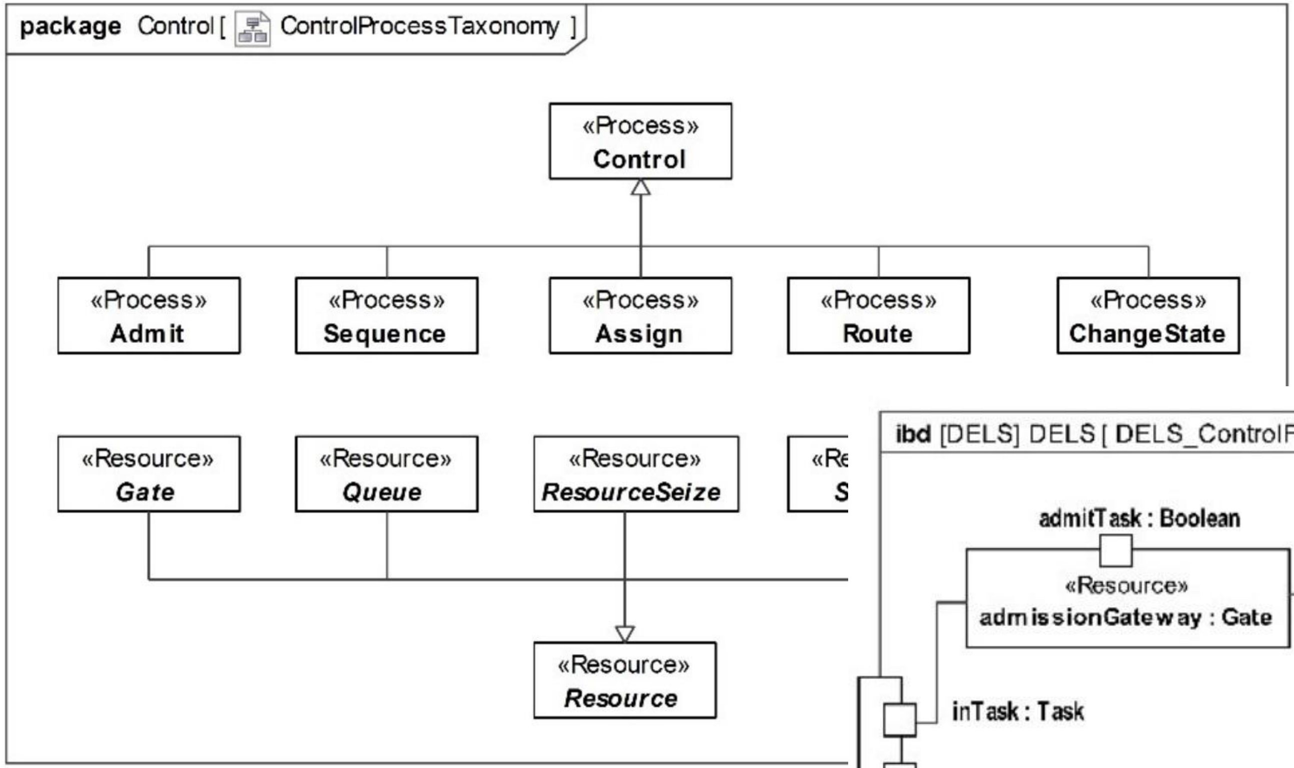
Operational Control

Functional mechanisms that **manipulate flows of tasks** and **resources** through a system in real-time, or near real-time.

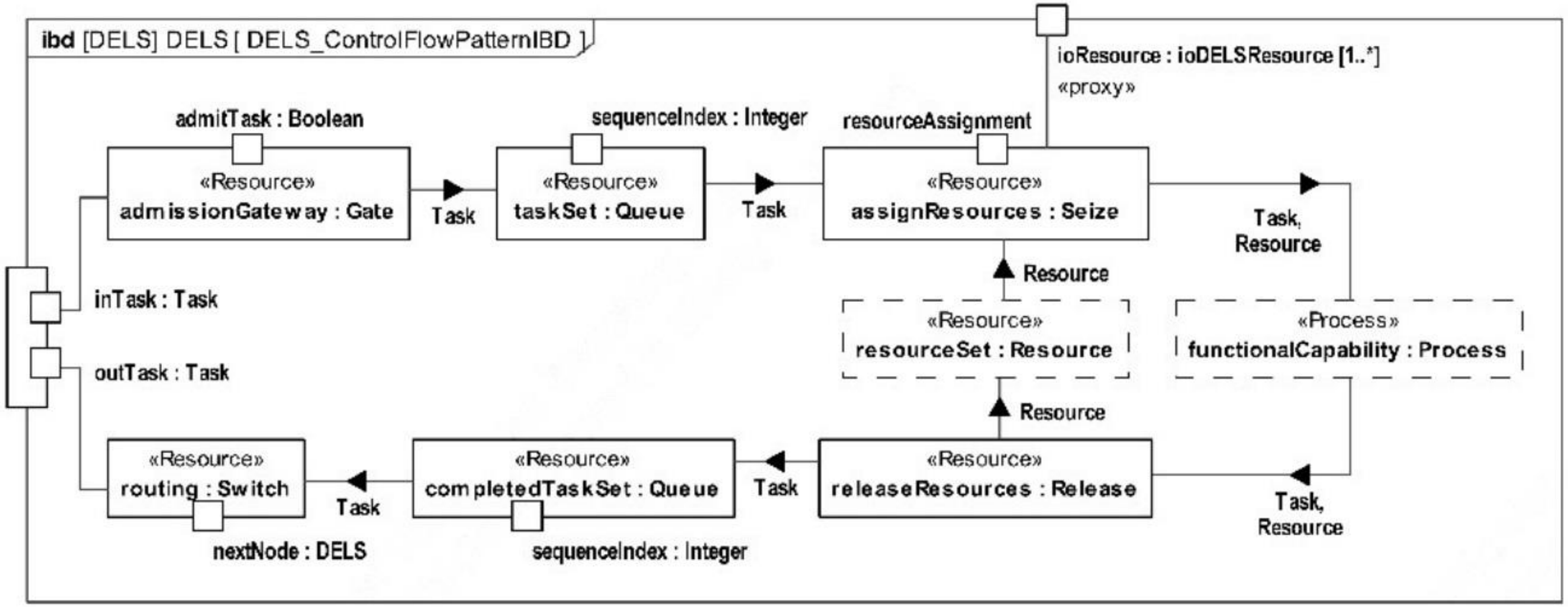


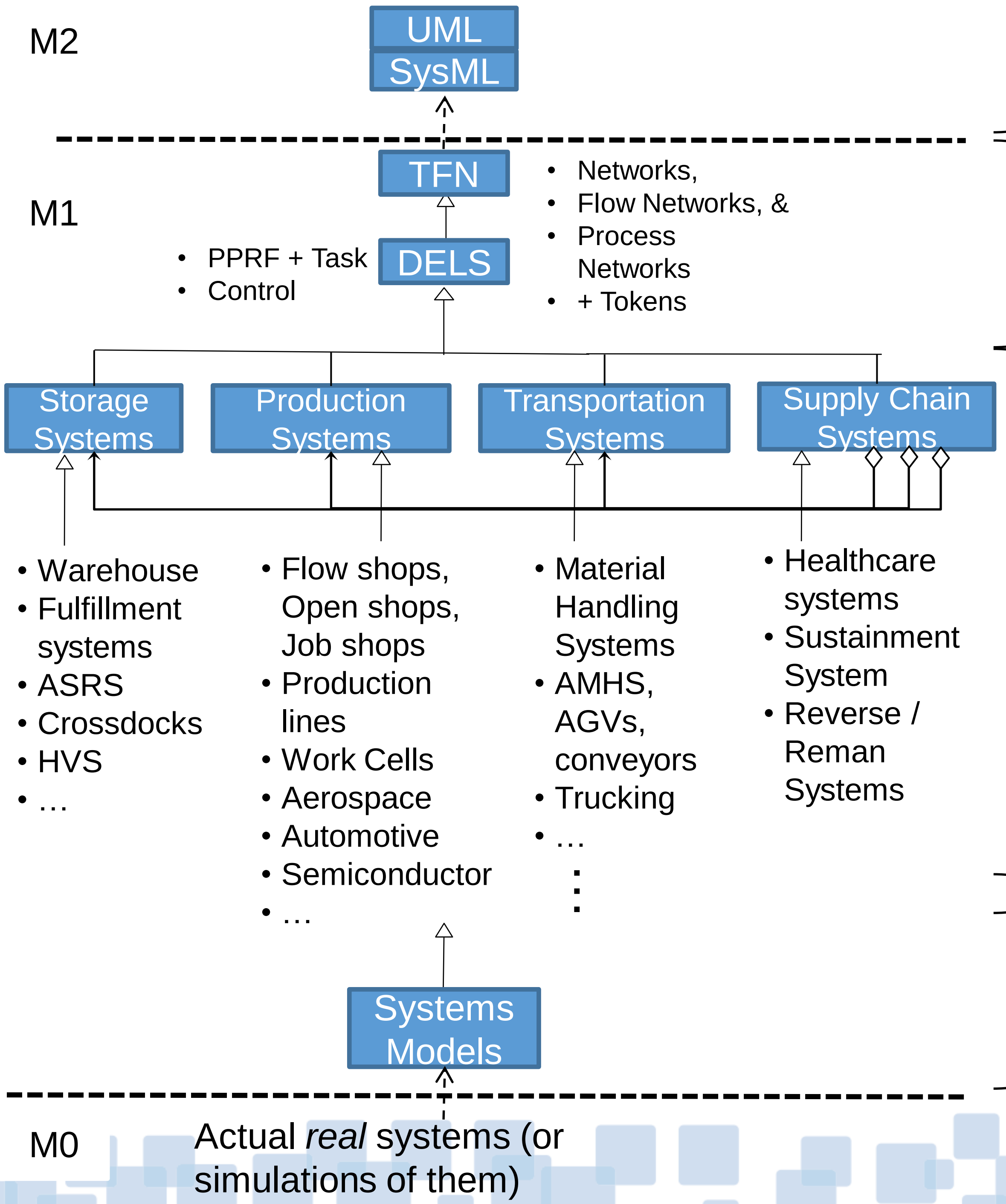
- Which tasks get serviced? (Admission/Induction)
- When {sequence, time} does a task get serviced? (Sequencing/Scheduling)
- Which resource services a task? (Assignment/Scheduling)
- Where does a task go after service? (Routing)
- What is the state of a resource? (task/services can it service/provide)

Operational Control



Maps the **decision variables** in the controller's decision problem to a particular **actuator function** and **execution mechanism** in the plant





<u>Language Layer</u>

- May also include a TFN & DELS DSL

<u>Top of M1</u>

- **DELS Reference model**
- Network Abstractions
- PPRF Domain Ontology
- PPRF Taxonomies & Model Libraries
- Control Patterns

<u>Middle of M1</u>

- **(sub-) Domain-specific reference models and architectures**
- Generalization Set aligns with STORE, MAKE, & MOVE processes

<u>Bottom of M1</u>

- **System Models**
- "as-built" or "specification" models

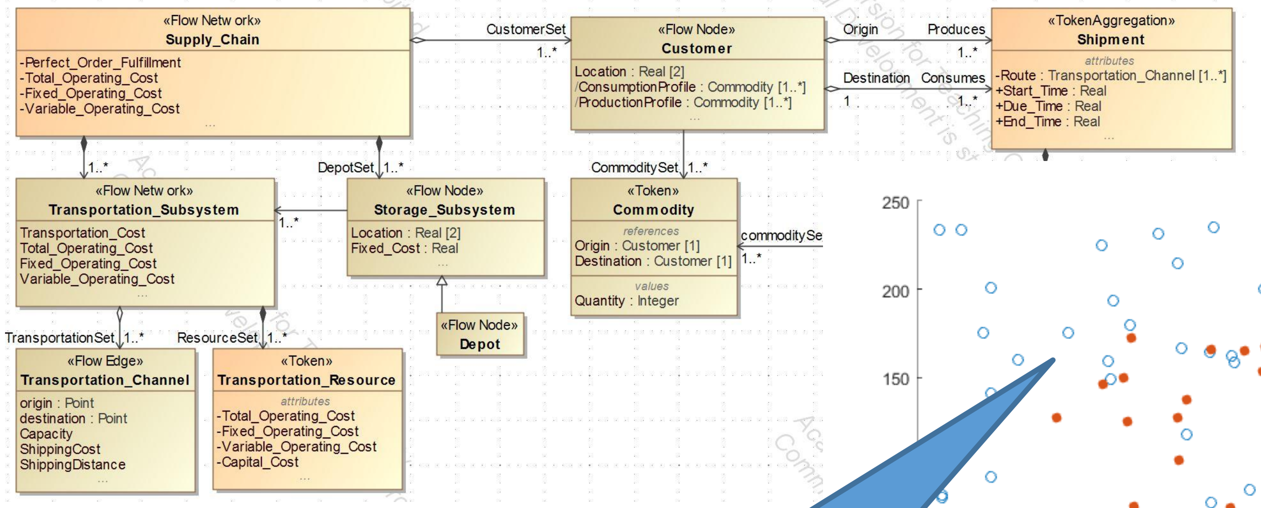
NIST
National Institute of Standards and Technology
U.S. Department of Commerce

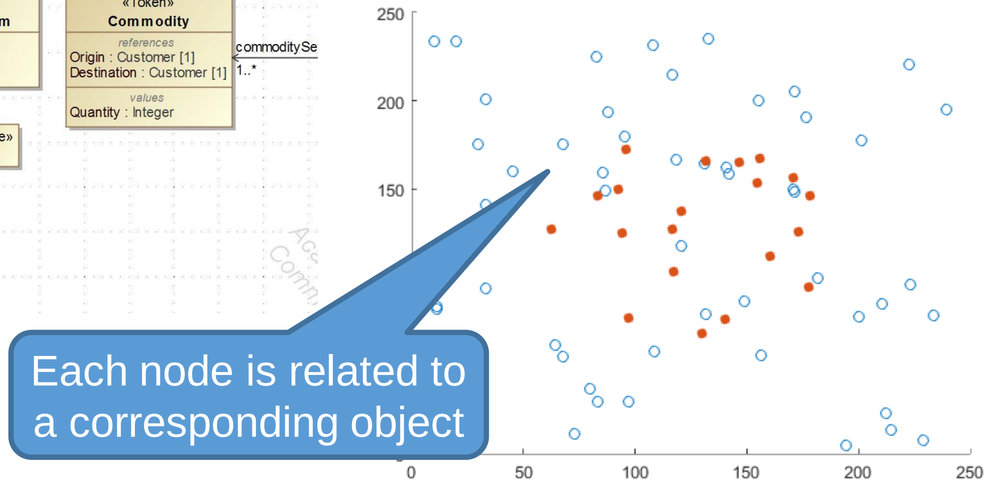
Georgia Tech

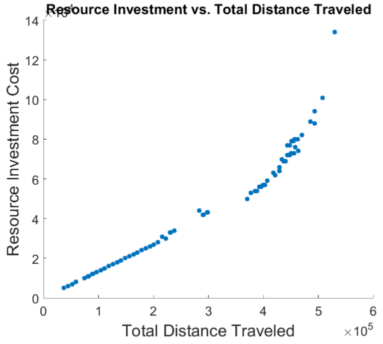
Outline

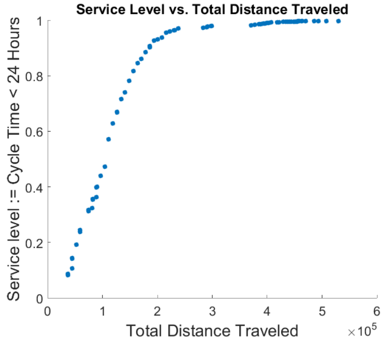
- What are DELS?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
  - What are the types of analysis models and problems we're interested in for DELS (SAI)?
- Where are we now?
  - What is contained in the DELS reference model?
  - **System-Analysis Integration Use Case**
- Where do we want to go?

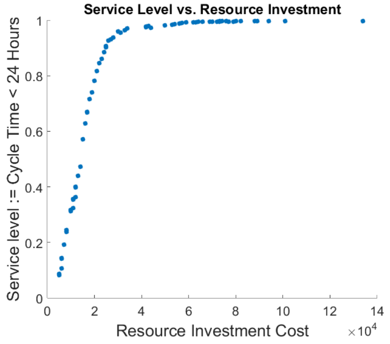
System-Analysis Integration – Use Case









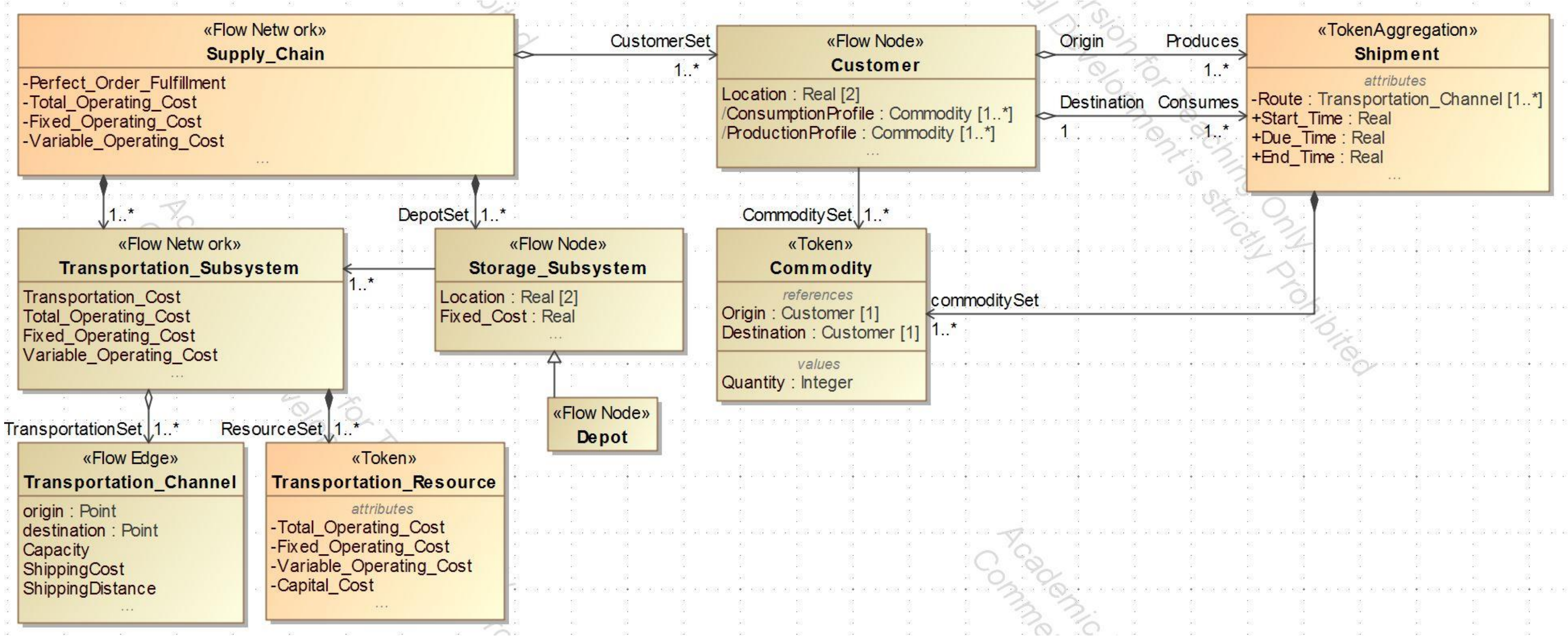


Strategy:

- Start with a system model or a reference model
- Generate an analysis model from the system model
- Use analysis model to support design decision making
- OR connect to an optimization model and search for candidate designs

Reference Models

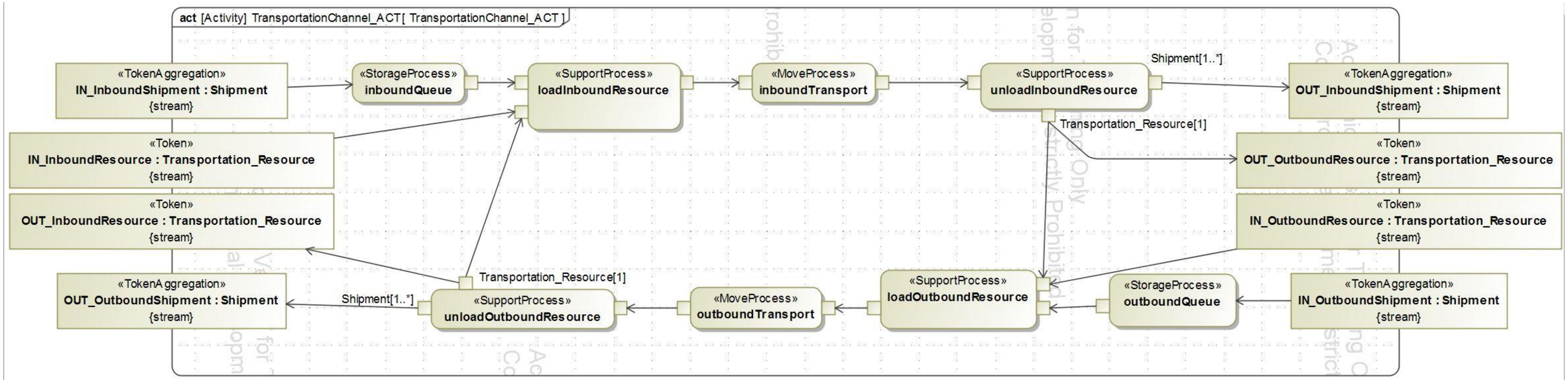
Domain-specific (supply chain) reference model provides a pattern for constructing conforming system instance models and analysis models.



➢ The system of interest for the use case presented is a distribution supply chain.

Transportation Channel Behavior

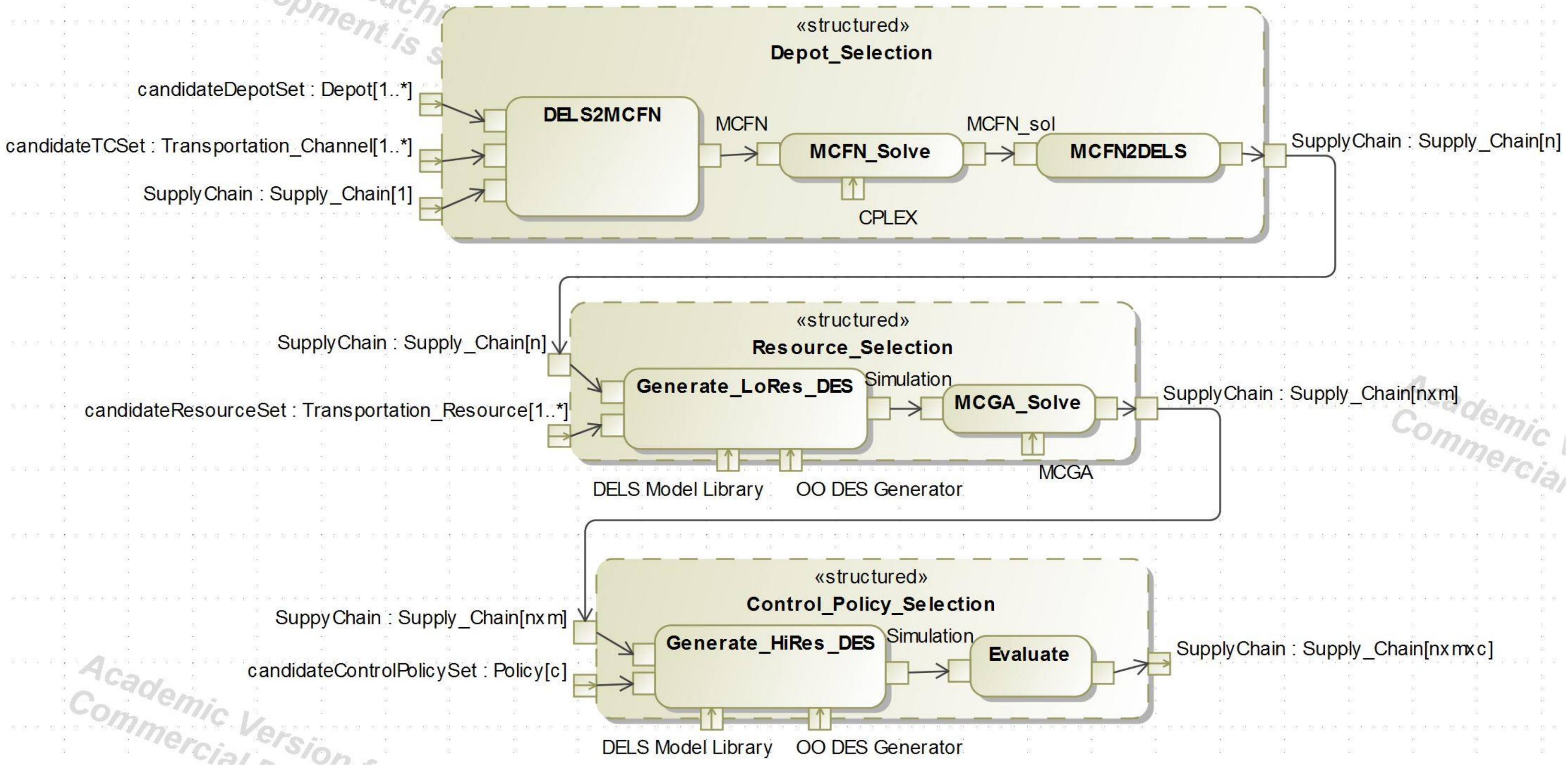
A formal specification of the behavior of the transportation channel provides a template for constructing the corresponding (simulation) analysis component.



Component-based generative methods for analysis models
V&V of components, compose models from model library components (bottom up)

Analysis Methodology overview

Hierarchical design methodology that uses tailored simulation optimization methods at each level to optimize the structure, behavior, and control of the DELS

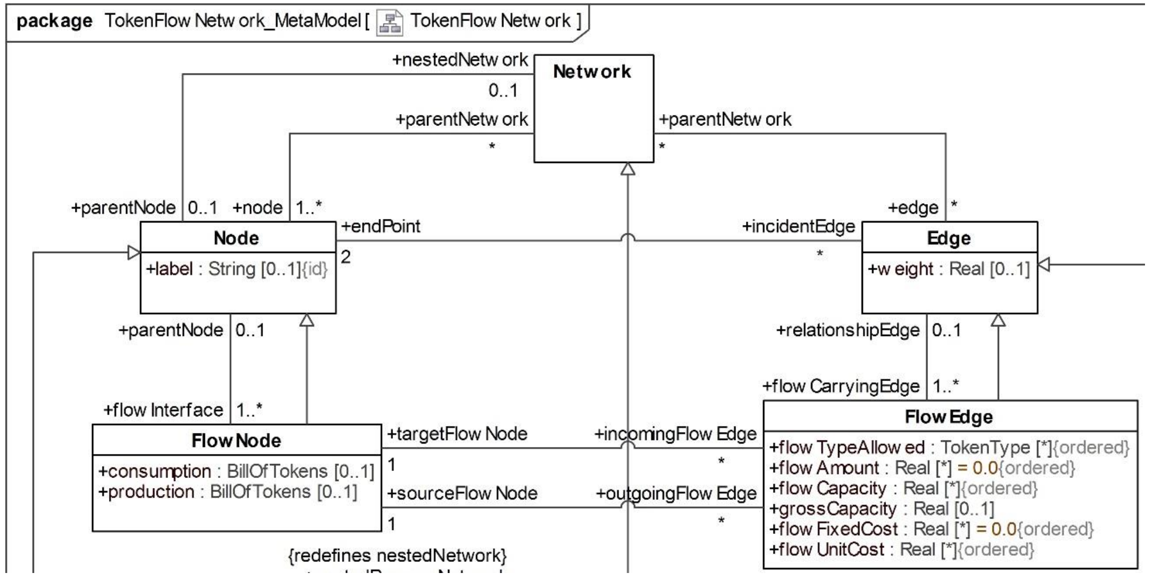


- Generate a large number of candidate solutions with corresponding simulation models specified at varying levels of aggregate, approximation, and resolution

NIST National Institute of Standards and Technology U.S. Department of Commerce

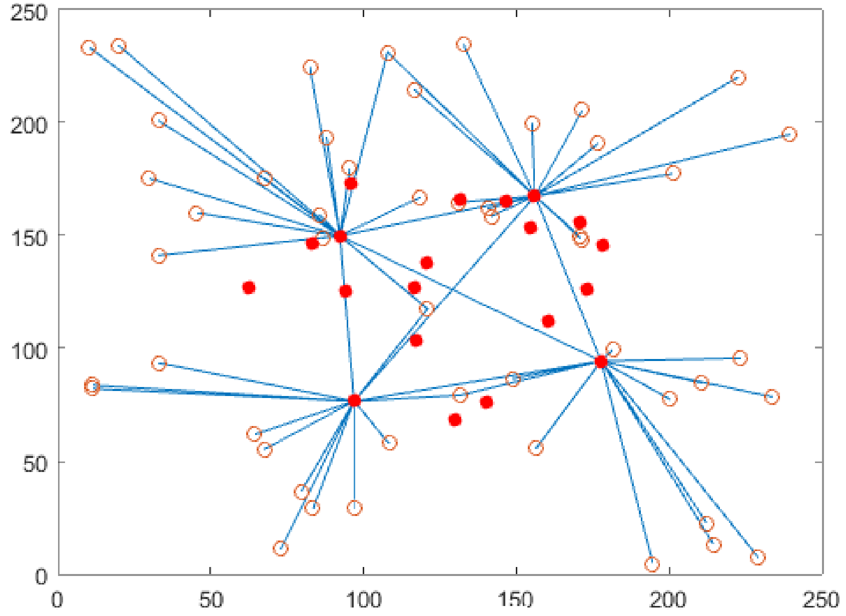
Georgia Tech

Structure: Depot Selection via MCFN



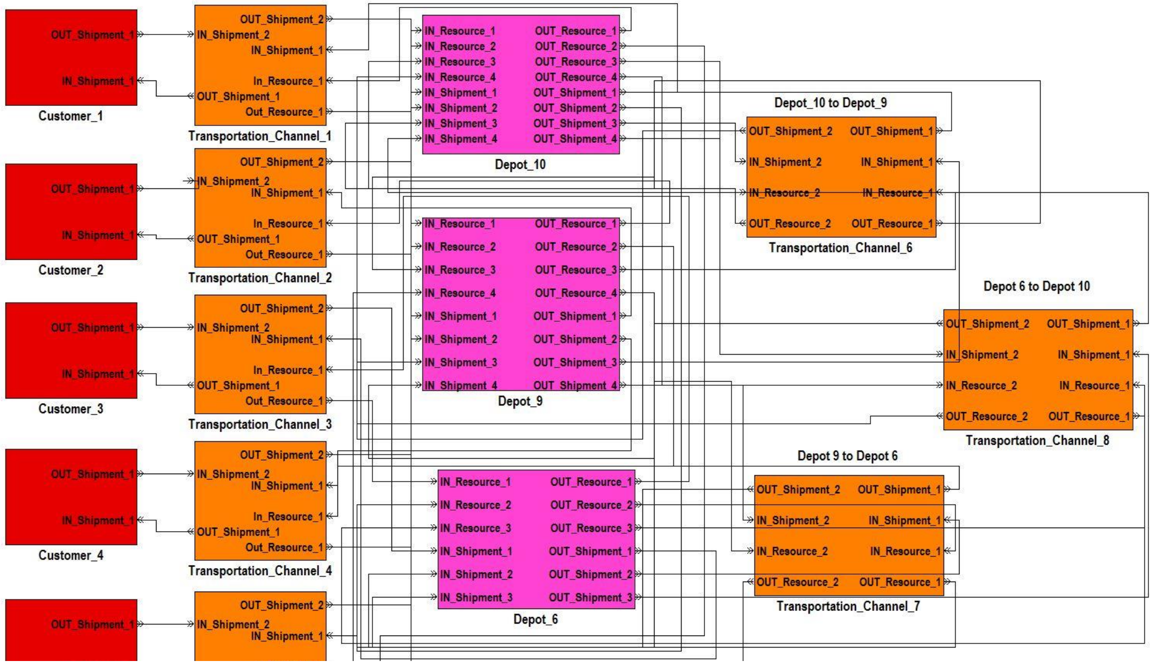
Goal: Reduce the computational requirements of optimizing the distribution network structure.

Strategy: Formulate and solve a corresponding multi-commodity flow network and facility location problem.



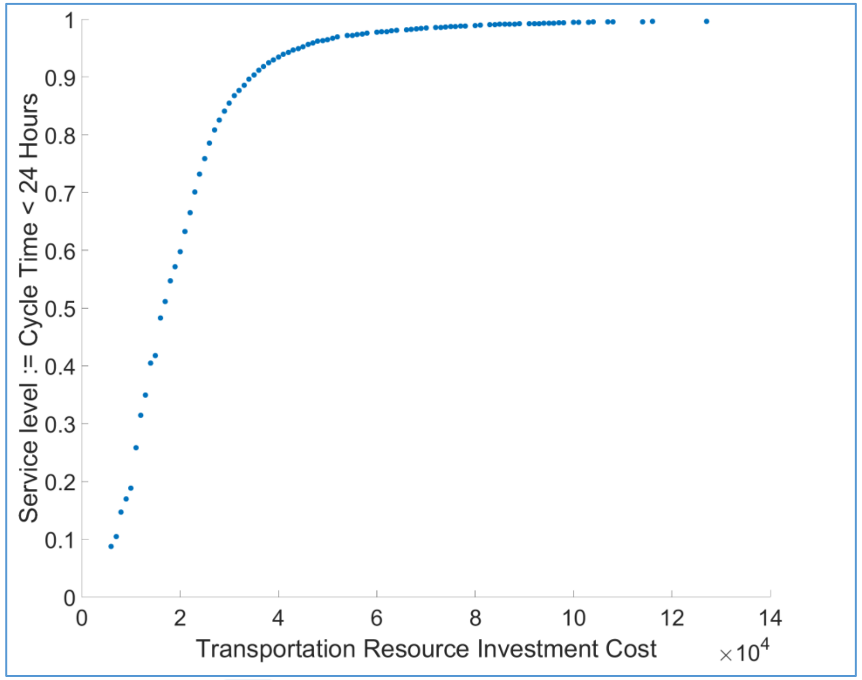
- Aggregate and approximate the flows and costs
- Solve MCFN using a COTS solver (CPLEX)
- Apply a "leave one out" strategy to generating several feasible candidate network structures.

Behavior: Resource Selection



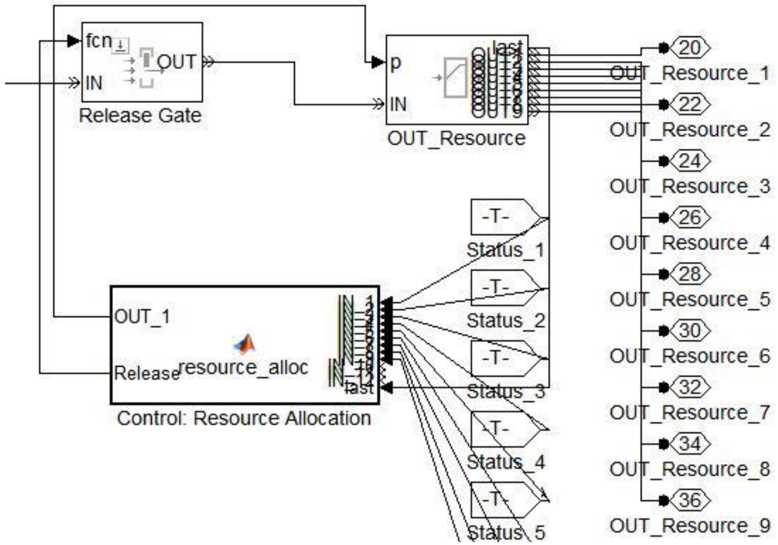
Goal: Capture and evaluate the behavioral aspects of the system using discrete event simulation.

Strategy: Generate a DES that simulates a probabilistic flow of commodities through the system.



- For each candidate supply chain network structure, generate a portfolio of solutions to the fleet sizing problem
- Trade-off cycle time/service level and resource investment cost

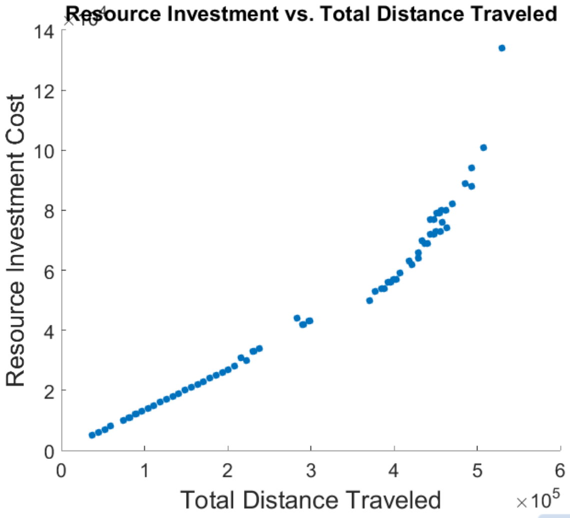
Control: Resource Assignment

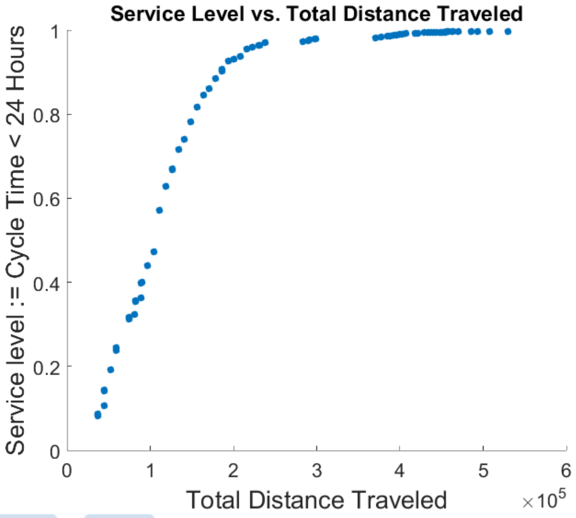


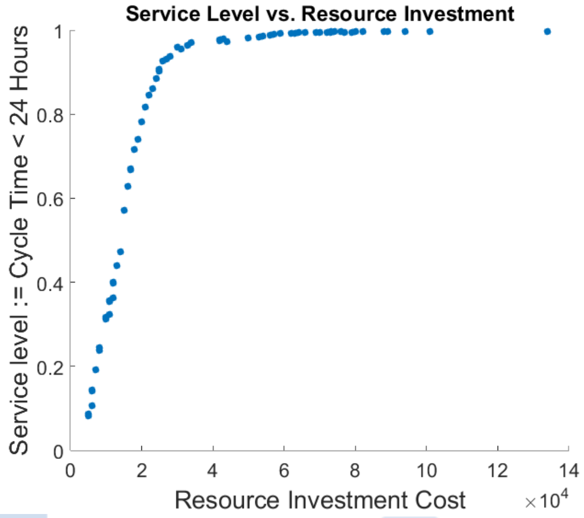
Goal: Select and design a detailed specification of the control policies for assigning trucks to pickup/dropoff tasks at customers.

Strategy: Generate a high-fidelity simulation that is detailed enough to fine-tune resource and control behavior.

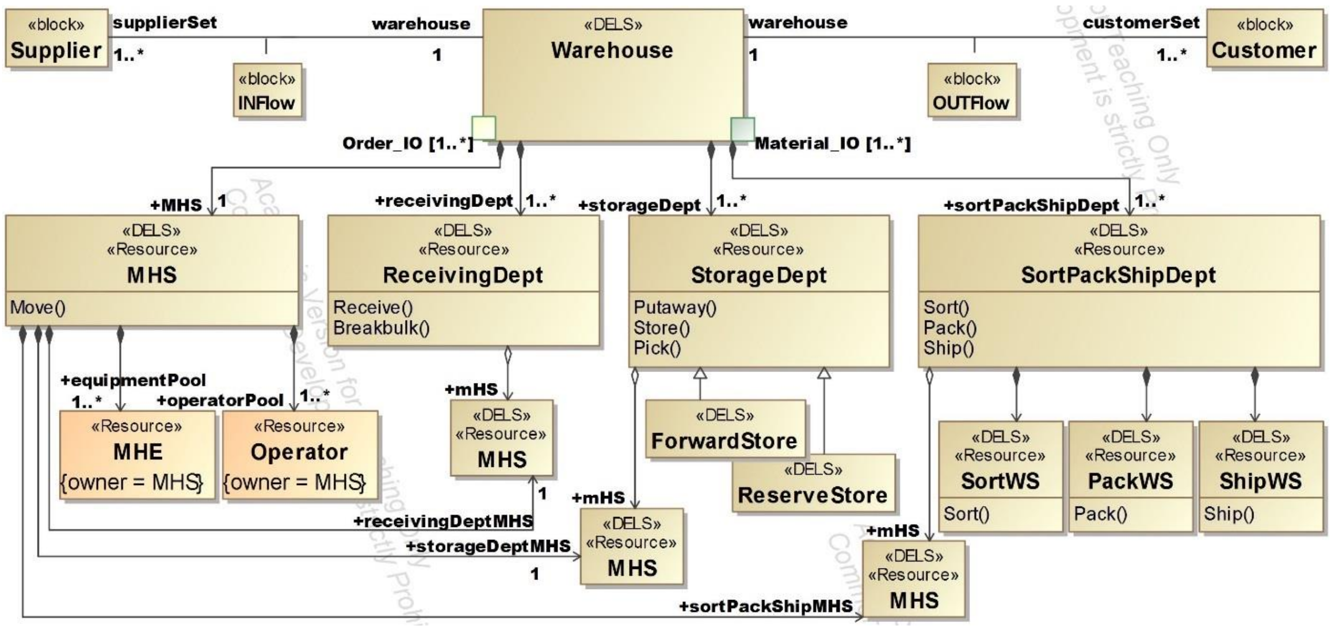
Trade-off Service Level, Capital Costs, and Travel Distance

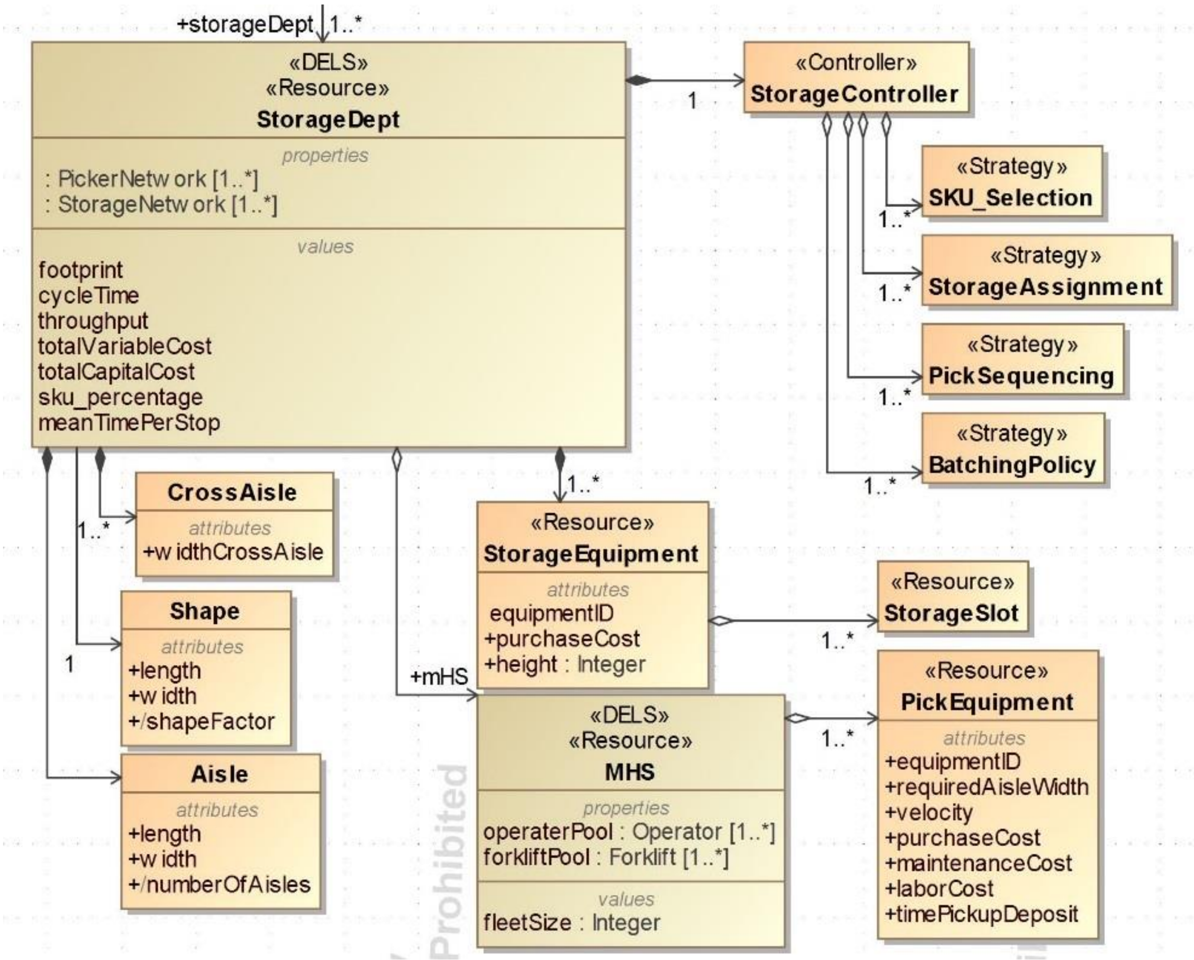




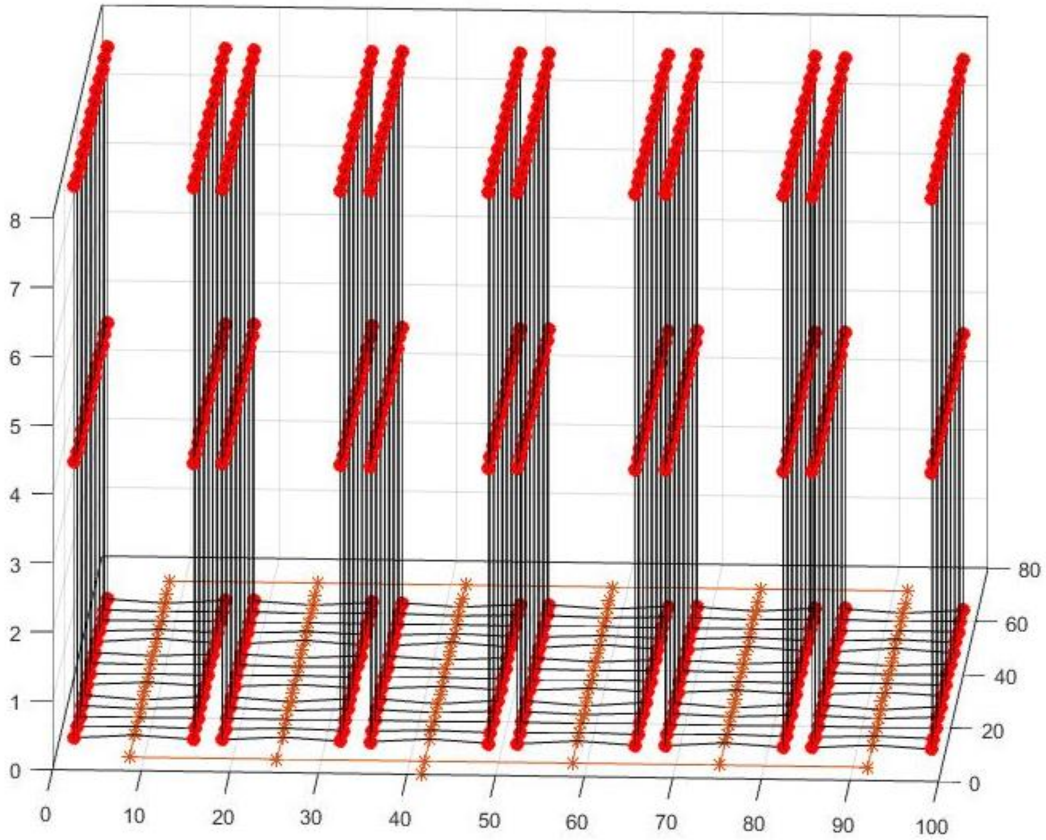


Warehouses


«block» Supplier
supplierSet
1..*
warehouse
1
«DELS» Warehouse
warehouse
1
customerSet
1..*
«block» Customer
«block» INFlow
«block» OUTFlow
Order_IO [1..*]
Material_IO [1..*]
+MHS 1
+receivingDept 1..*
+storageDept 1..*
+sortPackShipDept 1..*
«DELS» «Resource» MHS
Move()
«DELS» «Resource» ReceivingDept
Receive()
Breakbulk()
«DELS» «Resource» StorageDept
Putaway()
Store()
Pick()
«DELS» «Resource» SortPackShipDept
Sort()
Pack()
Ship()
+equipmentPool 1..*
+operatorPool 1..*
«Resource» MHE {owner = MHS}
«Resource» Operator {owner = MHS}
+mHS
«DELS» «Resource» MHS
«DELS» ForwardStore
«DELS» ReserveStore
«DELS» «Resource» SortWS
Sort()
«DELS» «Resource» PackWS
Pack()
«DELS» «Resource» ShipWS
Ship()
+receivingDeptMHS
+storageDeptMHS
+sortPackShipMHS

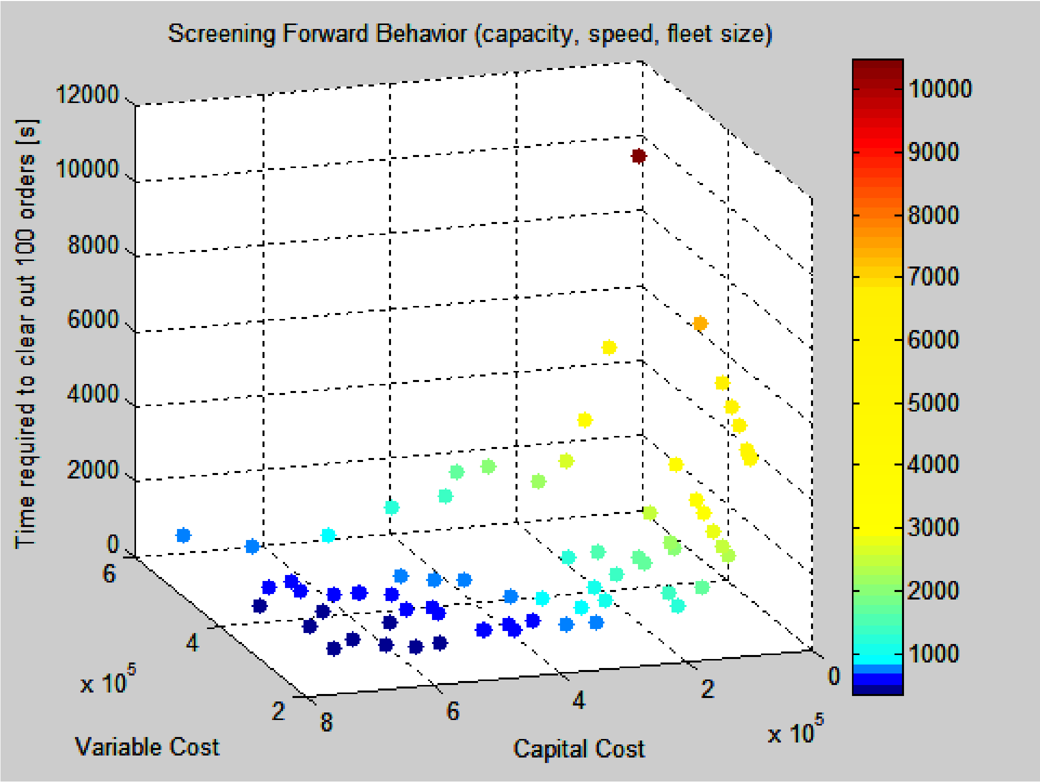

+storageDept 1..*
«DELS» «Resource» StorageDept
properties
: PickerNetwork [1..*]
: StorageNetwork [1..*]
values
footprint
cycleTime
throughput
totalVariableCost
totalCapitalCost
sku_percentage
meanTimePerStop
1
«Controller» StorageController
«Strategy» SKU_Selection
«Strategy» StorageAssignment
«Strategy» PickSequencing
«Strategy» BatchingPolicy
1..*
CrossAisle
attributes
+widthCrossAisle
Shape
attributes
+length
+width
+/shapeFactor
Aisle
attributes
+length
+width
+/numberOfAisles
«Resource» StorageEquipment
attributes
equipmentID
+purchaseCost
+height : Integer
«Resource» StorageSlot
+mHS
«DELS» «Resource» MHS
properties
operaterPool : Operator [1..*]
forkliftPool : Forklift [1..*]
values
fleetSize : Integer
«Resource» PickEquipment
attributes
+equipmentID
+requiredAisleWidth
+velocity
+purchaseCost
+maintenanceCost
+laborCost
+timePickupDeposit

Analysis Model Generation



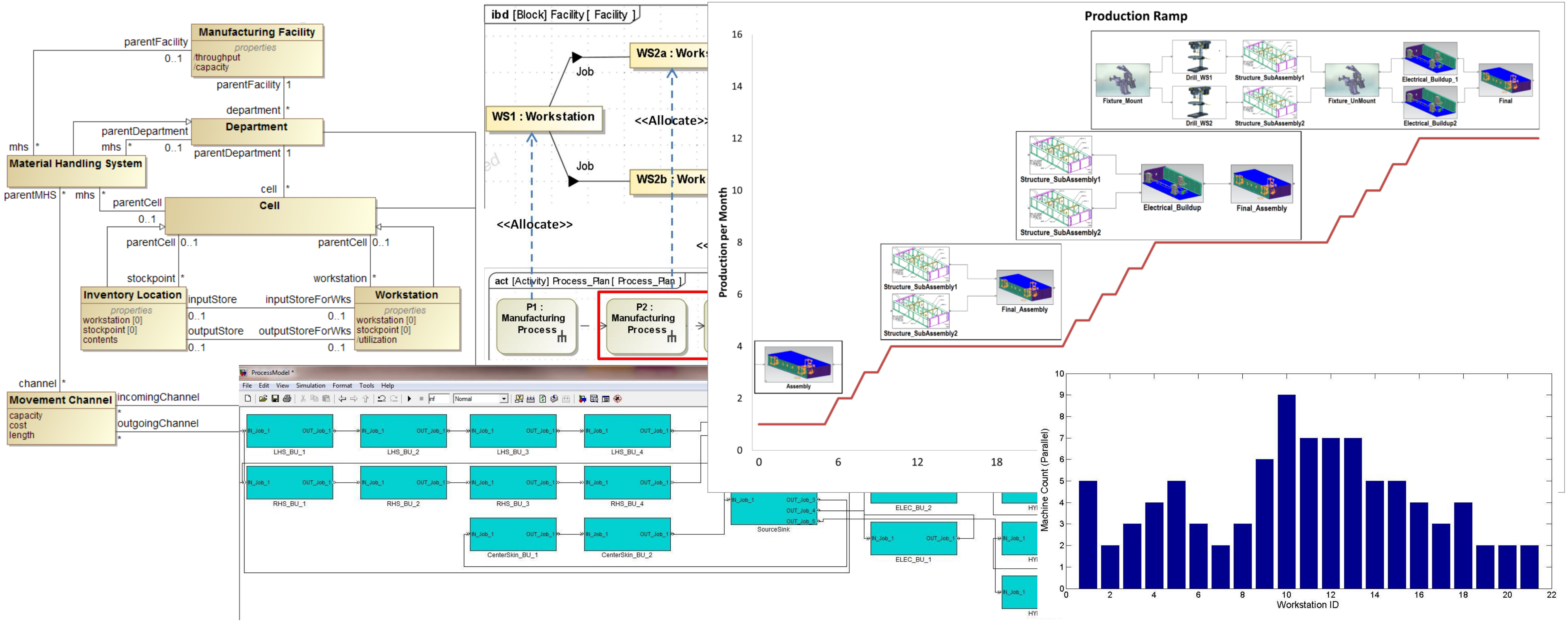
Metrics to support decision making:

- time required to clear out 100 orders (proxy for throughput),
- average time per tour (proxy for cycle time),
- capital cost,
- variable cost



For each layout, simulation model evaluates the performance of the storage and retrieval behavior and control

Manufacturing Facilities



NIST National Institute of Standards and Technology U.S. Department of Commerce

Georgia Tech

Why do it this way?

- The design and analysis of these systems tends to reuse common components
  - Extensible to support DELS domain
- Generate many simulation models from the system model at varying degrees of fidelity, aggregation, and approximation
- Interoperability based on a formal domain model allows tailoring of analysis methods to take advantage of domain-specific strategies.
  - Optimization heuristics
  - Advances in simulation and computing technology
  - Integrate with information systems for real-time data, providing decision-support, and executing operational control

Ongoing Work: What problem are we trying to solve?

- Mediate simulation and optimization tools with an explicit system model
  - A formal system model enables a greater degree of (semantic) interoperability

- Good progress so far on modeling the network structure and behavior of DELS
  - Structure: Networks, Flow Networks, & Process Networks
  - Behavior: Product, Process, Resource, & Facility models

- Leaves an unmet requirement for a formal specification and conforming analysis methods for operational control.

- What we need: An agreed upon and explicit definition of operational control for DELS

Where do we want to go?

- INCOSE MBSE Initiative WG on DELS Modeling
  - Single community for modeling DELS
    - Investigate crossover with transportation and healthcare WGs

- Connect to and engage with production system and logistics organizations
  - For every company that would like to see the benefits of MBSE in their manufacturing and supply chain organizations

For more information

leon.mcginnis@isye.gatech.edu

timothy.sprock@nist.gov

conrad.bock@nist.gov

2017

annual INCOSE

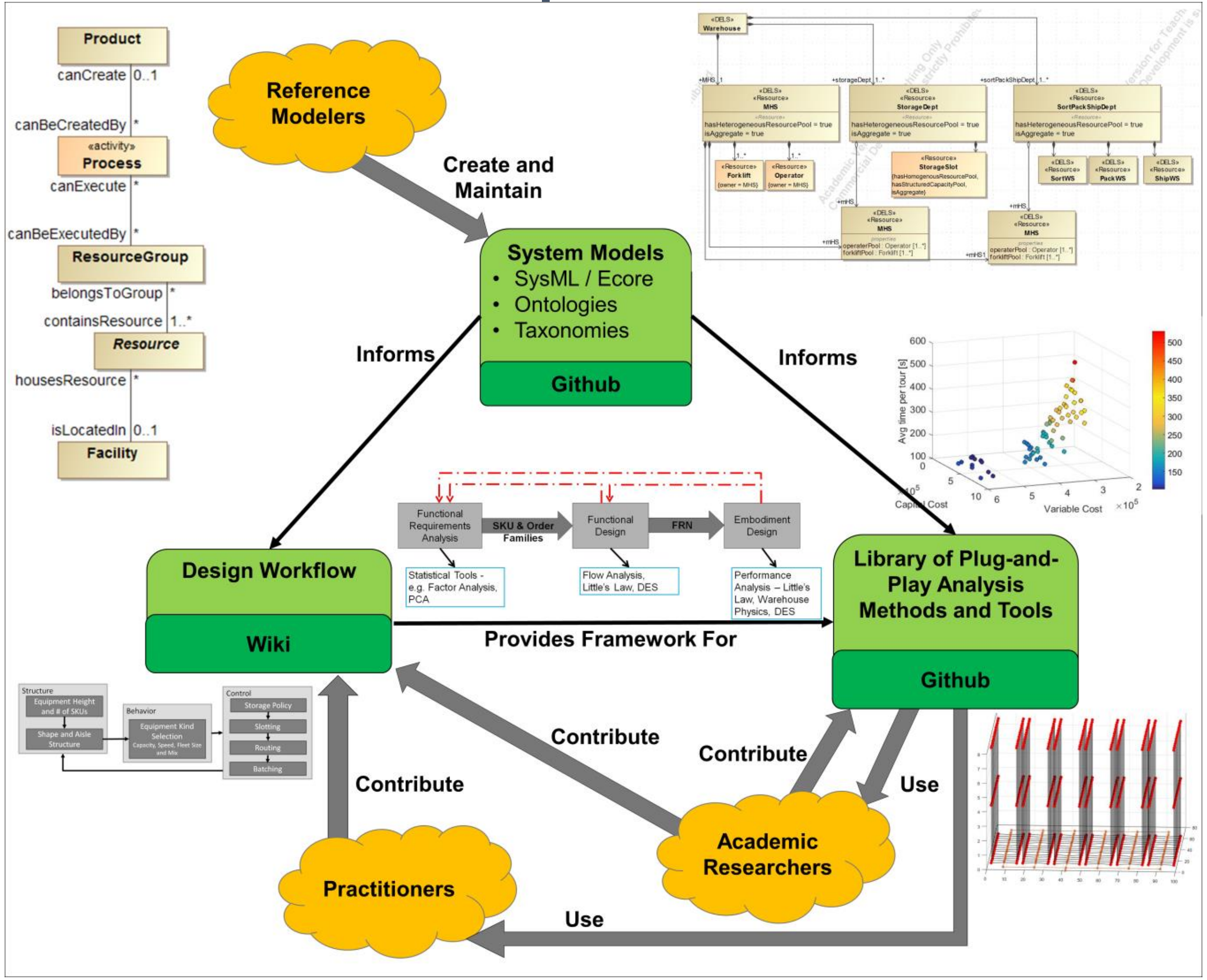
international workshop

Los Angeles, CA, USA

January 28 - 31, 2017

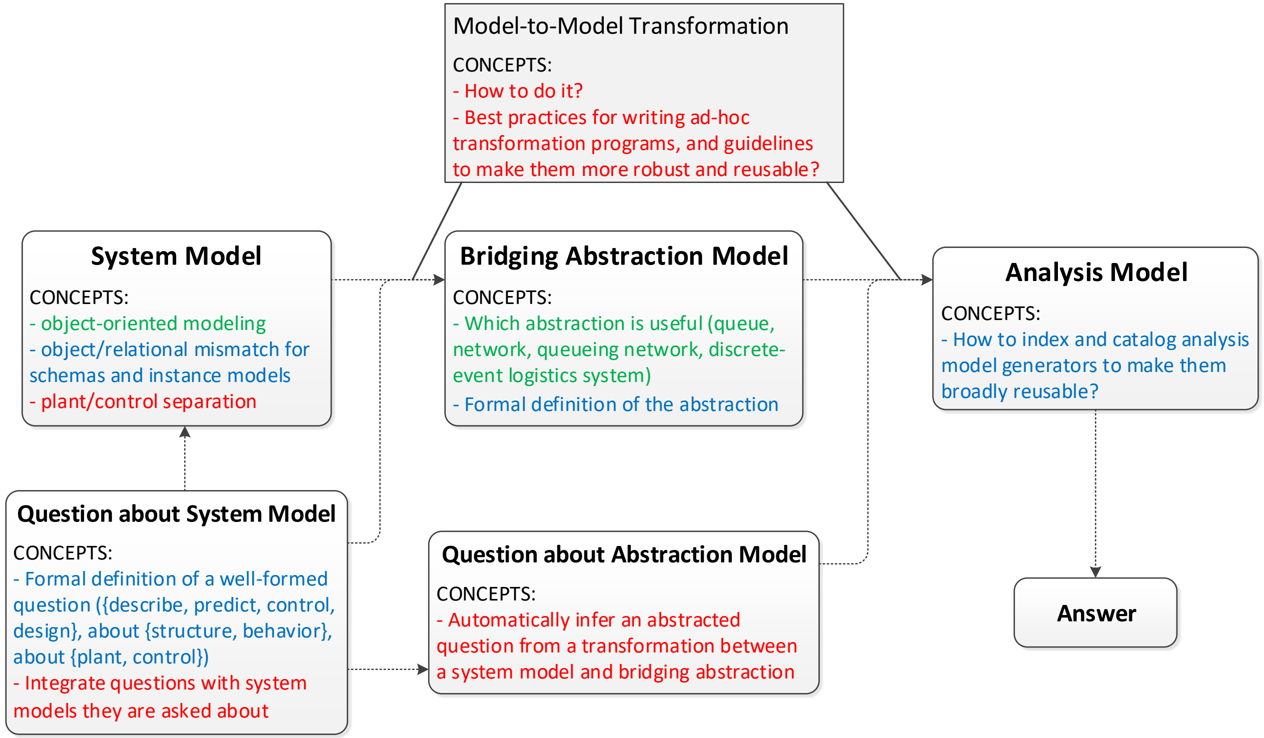
www.incose.org/IW2017

Mechanisms for development collaboration



National Institute of Standards and Technology
U.S. Department of Commerce
Georgia Tech

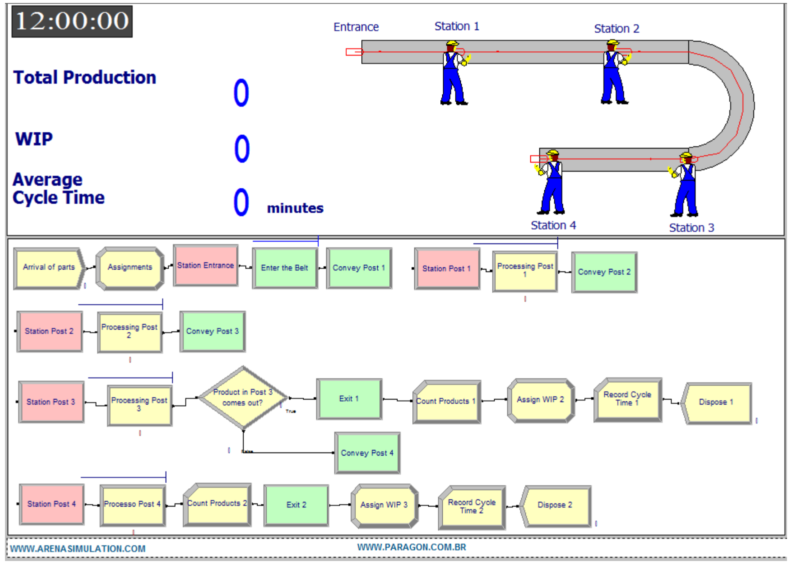
What are DELS? And Research Goals

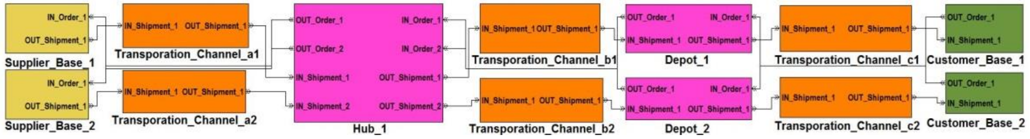


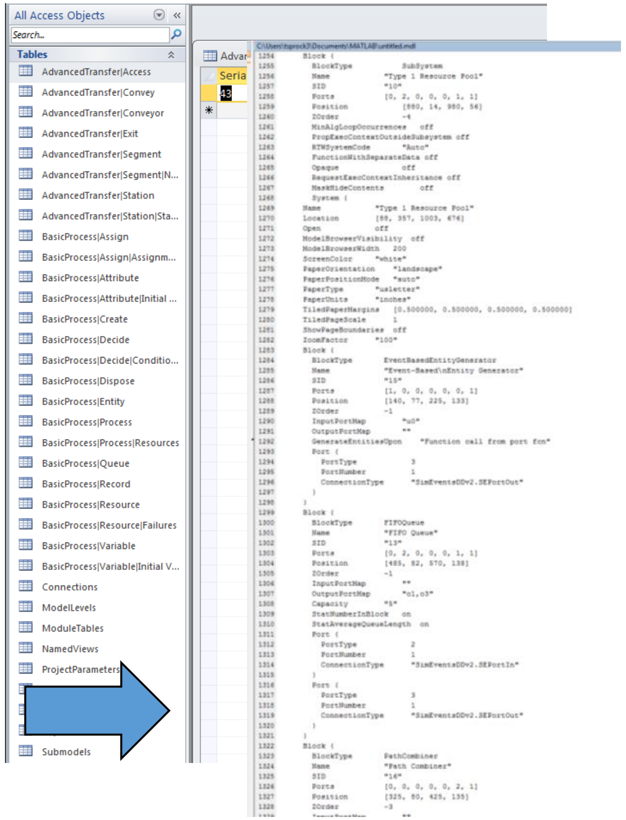
Domain Specific Challenges

Difficulties arise in applying current M2M methodologies for code generation to generating discrete event simulation.

Many popular simulation tools fail to store their models in a well-structured and accessible format, for which there is a published schema.



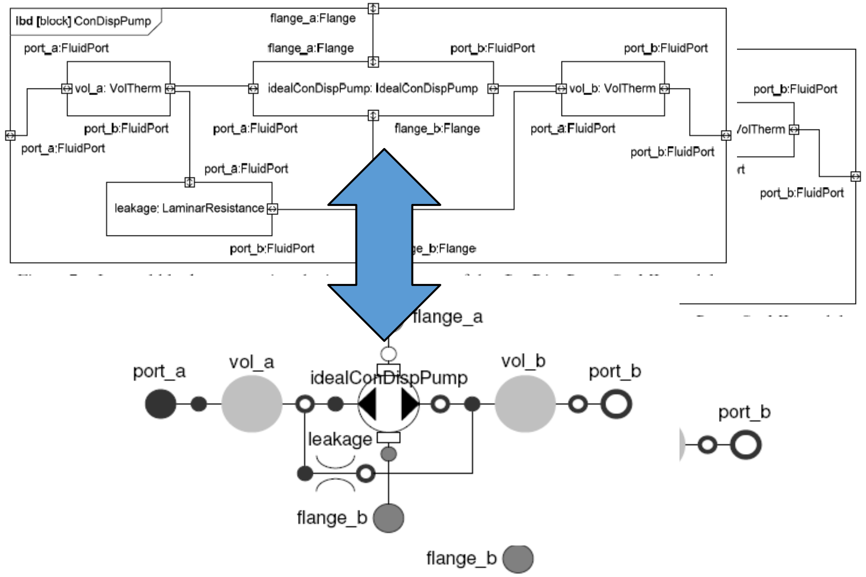




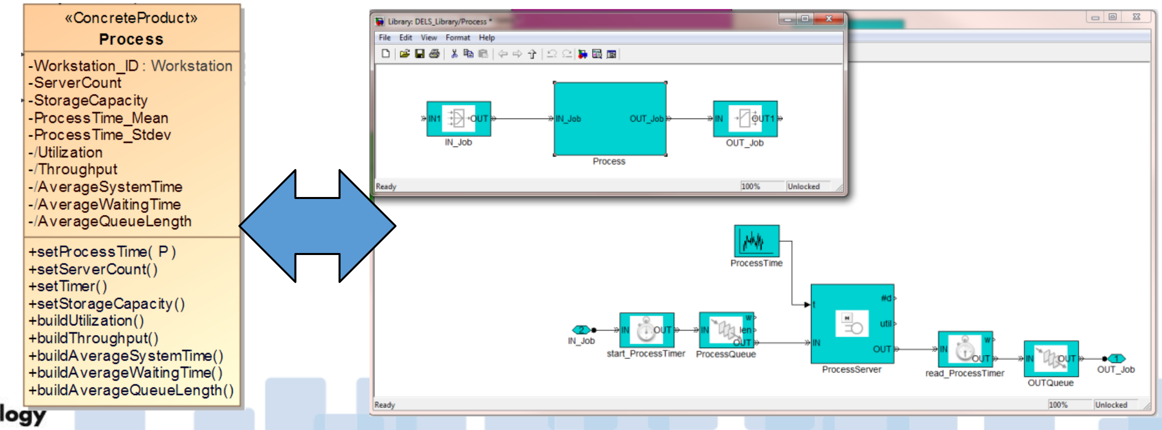
Similar issues with Tecnomatix PlantSim, FlexSim, etc.

Why is Discrete Event Simulation Hard?

OMG's SysML-Modelica Transformation (SyM), Version 1.0

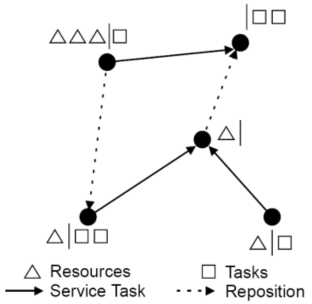


COTS Discrete Event Simulation languages lack a common abstraction and implementation



Discrete Optimization has a canonical set-based abstraction (Thiers, 2014)

- Queues of tasks and resources at each location in the network
- Resources move to service a task or reposition
- Tasks must be serviced within time window to receive full revenue
- Over time, new tasks arrive, old tasks expire



Physical Depiction of Dynamic Resource Allocation Problem

$$\min \quad c^T x$$

$$s.t. \quad Ax = b$$

$$x \geq 0$$

```
set FlowNode;
set FlowEdge within (FlowNode cross FlowNode);
set TokenType;

#Sign convention for netFlow:  Demand is positive, Supply is negative
param netFlow {FlowNode, TokenType};
param flowUnitCost {FlowEdge, TokenType};
param typeCapacity {FlowEdge, TokenType};
param grossCapacity {FlowEdge};

var flowAmount {FlowEdge, TokenType};

minimize netFlowCost:
sum {(i,j) in FlowEdge, c in TokenType}
flowUnitCost[(i,j),c] * flowAmount[(i,j),c];

subject to flowBalance {n in FlowNode, c in TokenType}:
sum {(i,n) in FlowEdge} flowAmount[(i,n),c]
   = netFlow[n,c] + sum{(n,j) in FlowEdge} flowAmount[(n,j),c];

subject to flowBounds {(i,j) in FlowEdge, c in TokenType}:
0 <= flowAmount[(i,j),c] <= typeCapacity[(i,j),c];

subject to edgeGrossCapacity {(i,j) in FlowEdge}:
sum {c in TokenType} flowAmount[(i,j),c] <= grossCapacity[(i,j)];
```