Integrating Systems Modeling with Engineering Analysis

Conrad Bock & Timothy Sprock

National Institute of Standards and Technology

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

Overview

- Project organization
- Motivation and approach
- Areas for integration
- Summary

Overview

- Project organization
- Motivation and approach
- Areas for integration
- Summary

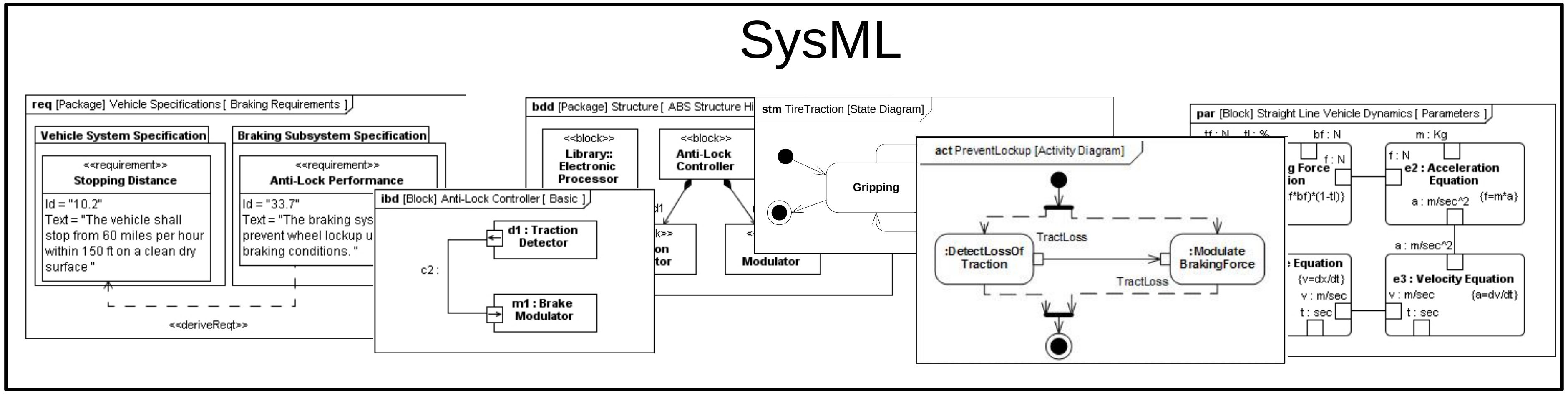
Systems Analysis Integration

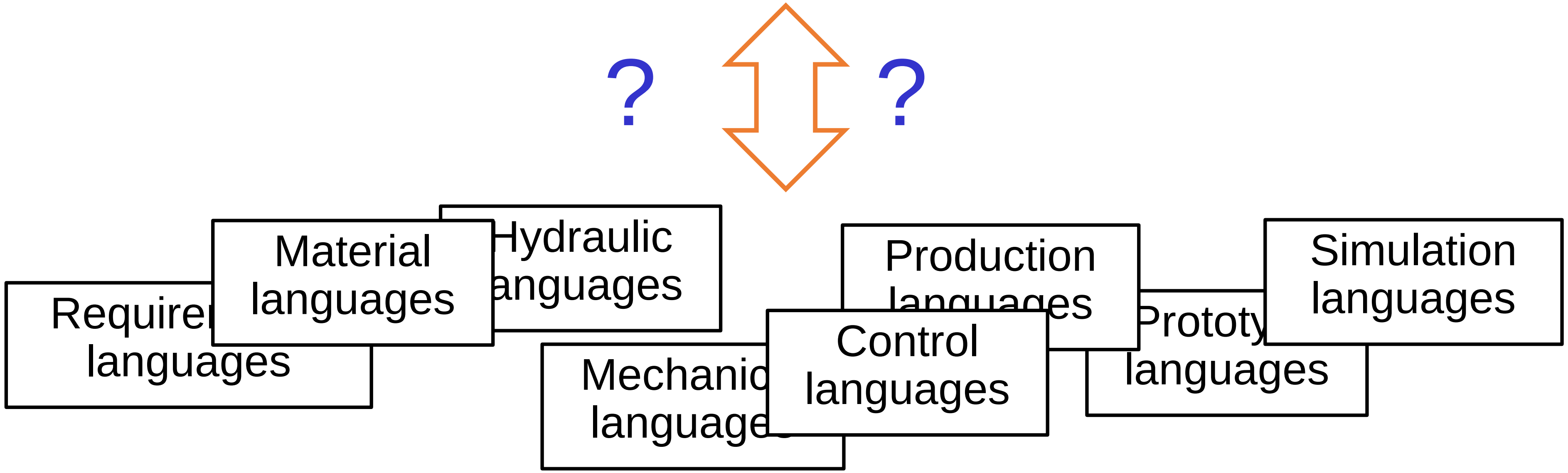
- Started later 2014.
- Concerned with integrating systems and other engineering models.
  - Will focus on integration of systems models and engineering analysis information.
- Two & ½ full time federal employees.
- Four multi-year cooperative agreements with universities and industry.

Overview

- Project organization
- Motivation and approach
- Areas for integration
- Summary

Engineering Language Integration





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

General Technical Approach

1. Select analysis areas to address.
2. Examine the literature and widely-used tools in those areas.
3. Develop information abstractions.
4. Identify overlap with SysML concepts.
   – Additional concepts for analyzing SE models.
5. Develop or choose integration technique.
6. Apply technique to SE/analysis gap.
7. Develop proof-of-concept.

Outputs

- All public domain.
- Standards (see next).
- Journal and other papers.
- Proofs of concept (open source).
- Presentations and demonstrations.

Overview

- Project organization
- Motivation and approach
- Areas for integration
- Summary

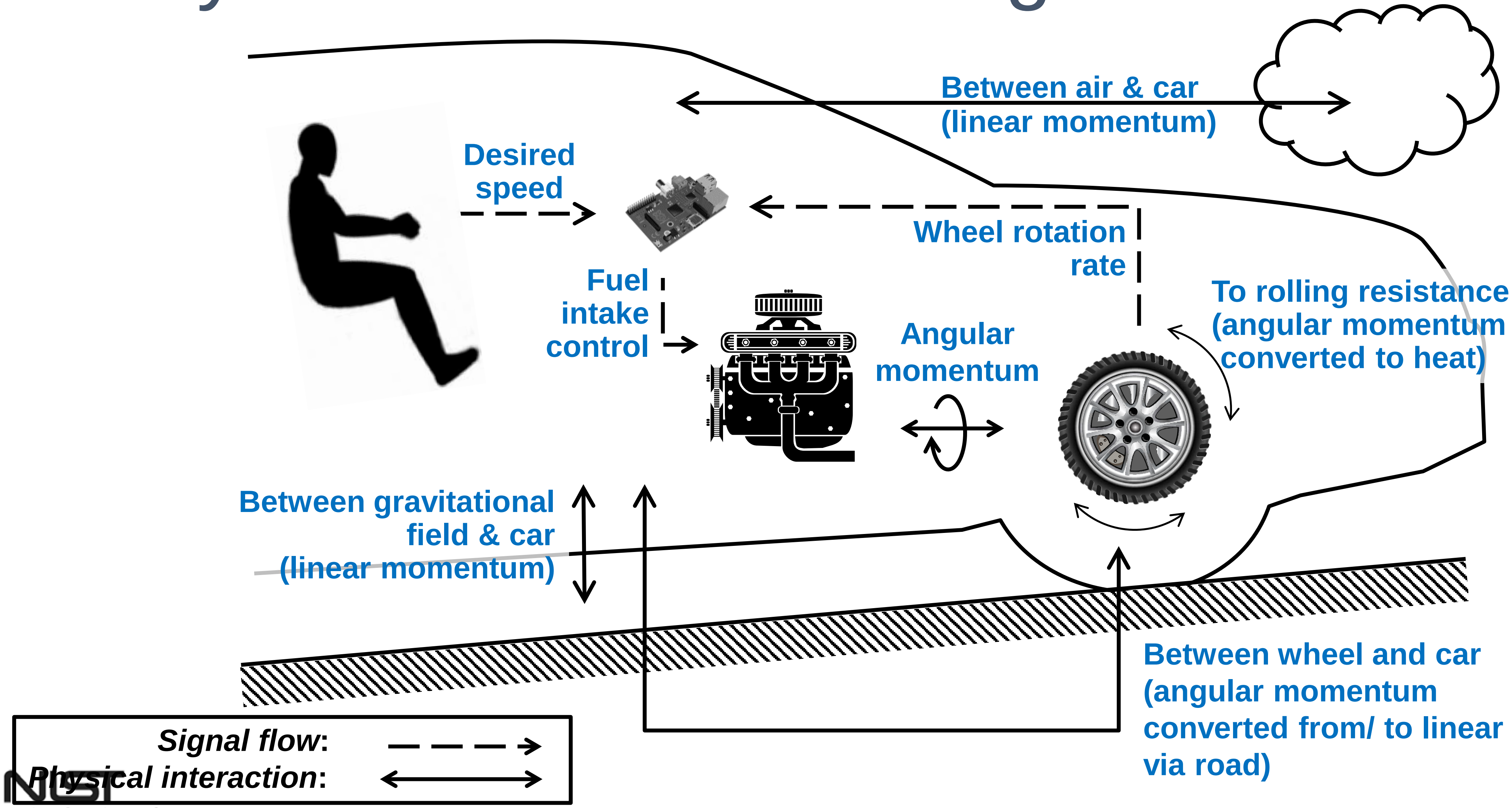
Areas for Integration

1. Physical interaction / signal flow simulation.
2. Finite element analysis.
3. Mathematical unification of systems and analysis models.
4. Discrete event analysis and optimization (production, logistics).
5. Sustainability analysis.

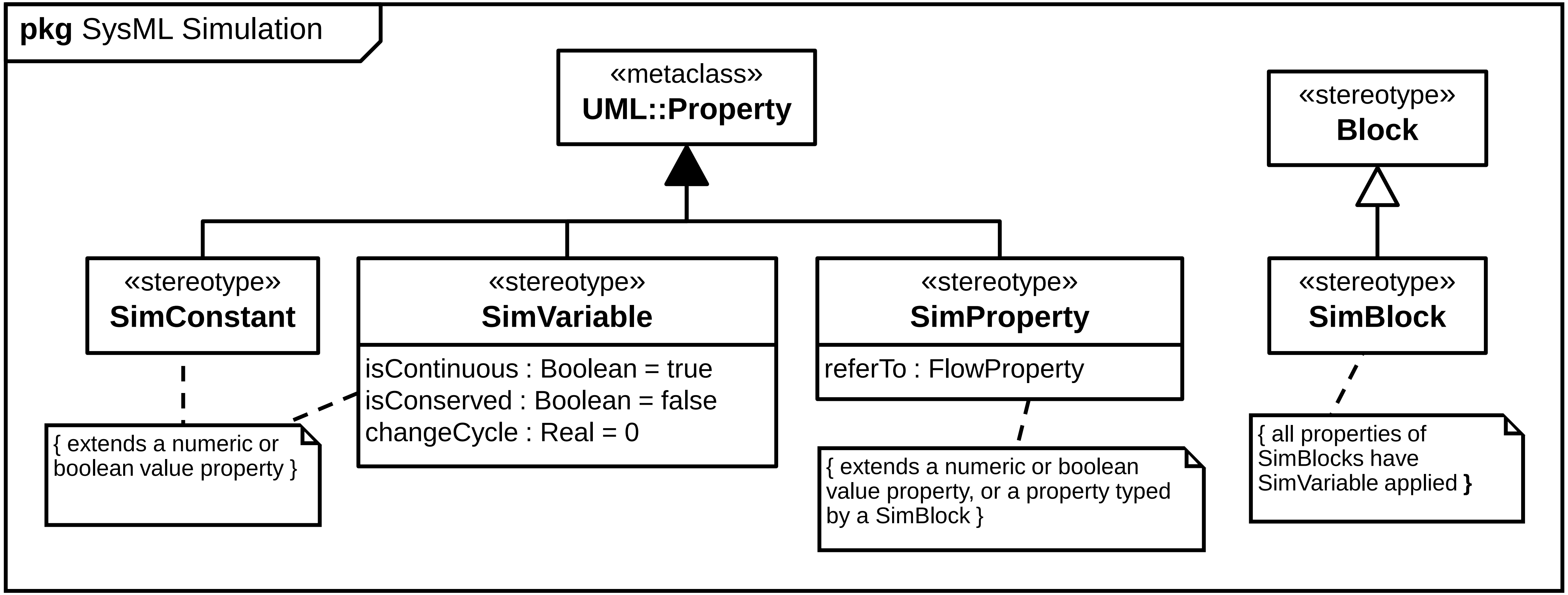
Areas for Integration

1. Physical interaction / signal flow simulation.
2. Finite element analysis.
3. Mathematical unification of systems and analysis models.
4. Discrete event analysis and optimization (production, logistics)
5. Sustainability analysis.

Physical Interaction & Signal Flow

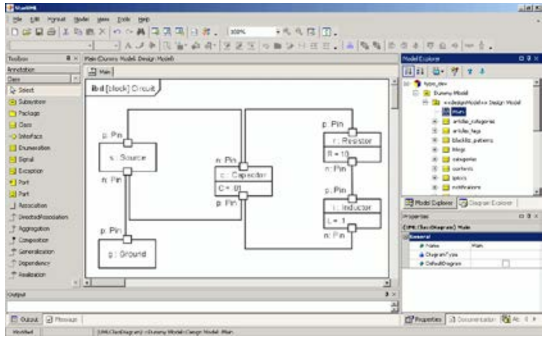


SysML Extension for PI & SF



- In finalization at the Object Management Group

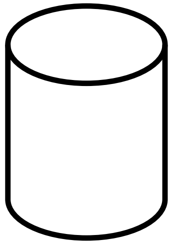
Open Source Translator to Simulators



SysML Tool

```
<xmi:XMI xmi:uml=http://www.omg.org/spec/UML/20110701>
<packagedElement xmi:type="uml:Class" name="Circuit">
  <ownedAttribute xmi:type="uml:Property” name="r"
   type="_17_0_2_1_8340219_1386362605823_957929_2536“/>
  <ownedAttribute xmi:type-"uml:Property” name="c"
   type="_17_0_2_1_8340219_1386362584651_224589_2534“/>
  <owned.Attr~bute xm1:type="uml:Property" name="i"
   type="_17_0_2_1_8340219_1386362591582_80360_2535“/>
  <ownedConnector xmi:type-"uml:Connector“/>
  <ownedConnector xmi:type-"uml:Connector“/>
  <ownedConnector xmi: type-"uml :Connector“/>
  <ownedConnector xmi:type-"uml:Connector“/>
</packagedElement>
<packagedElement xmi:type-"uml:Class" name="Source“/>
<packagedElement xmi:type-"uml:Class" name="Ground“/>
<packagedElement xmi:type-"uml:Class" name="Capacitor“/>
<packagedElement xmi:type-"um1:Association“/>
<packagedElement xmi:type="um1:Assoc1ation“/>
</uml:Model>
</xmi>
```

SysML Model File



UML Repository

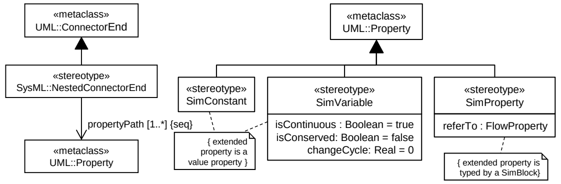
```
model circuit
  Resistor  R1(R=10);
  Capacitor C(C=0.01);
  Resistor  R2(R=100);
  Inductor  L(L=0.1);
  VsourceAC AC;
  Ground    G;
  equation
    connect (AC.p, R1.p);
    connect (R1.n, C.p);
    connect (C.n, AC.n);
    connect (R1.p, R2.p);
    connect (R2.n, L.p);
    connect (L.n,  C.n);
    connect (AC.n, G.p);
end circuit;
```

Modelica and Mathworks

Simulator Input File

loaded into

developed in

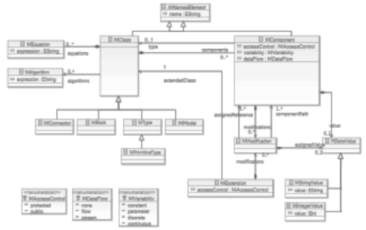


SysML and Simulator Extensions

```
public void generate(Class urootblock, Resource r) {
  sysmlutil = new SysMLUtil(r.getResourceSet());
  slibrary = SimulinkFactory.eINSTANCE.createSLib();
  slibrary.setName(urootblock.getName() + "Library");
  while(toProcess.size()!=0)
  { ReferenceKey rk = toProcess.pop();
    SElement selement = refs.get(rk);
    if (selement instanceof SSystem)
    { SSystem ssystem2 = processSys (rk.getKey()[0]);
      slibrary.getSystem().add(ssystem2.getSubsys());}
    else if (selement instanceof SFunction)
    { processSFunc(rk.getKey()[0], rk.getKey()[1);}
  }}
```

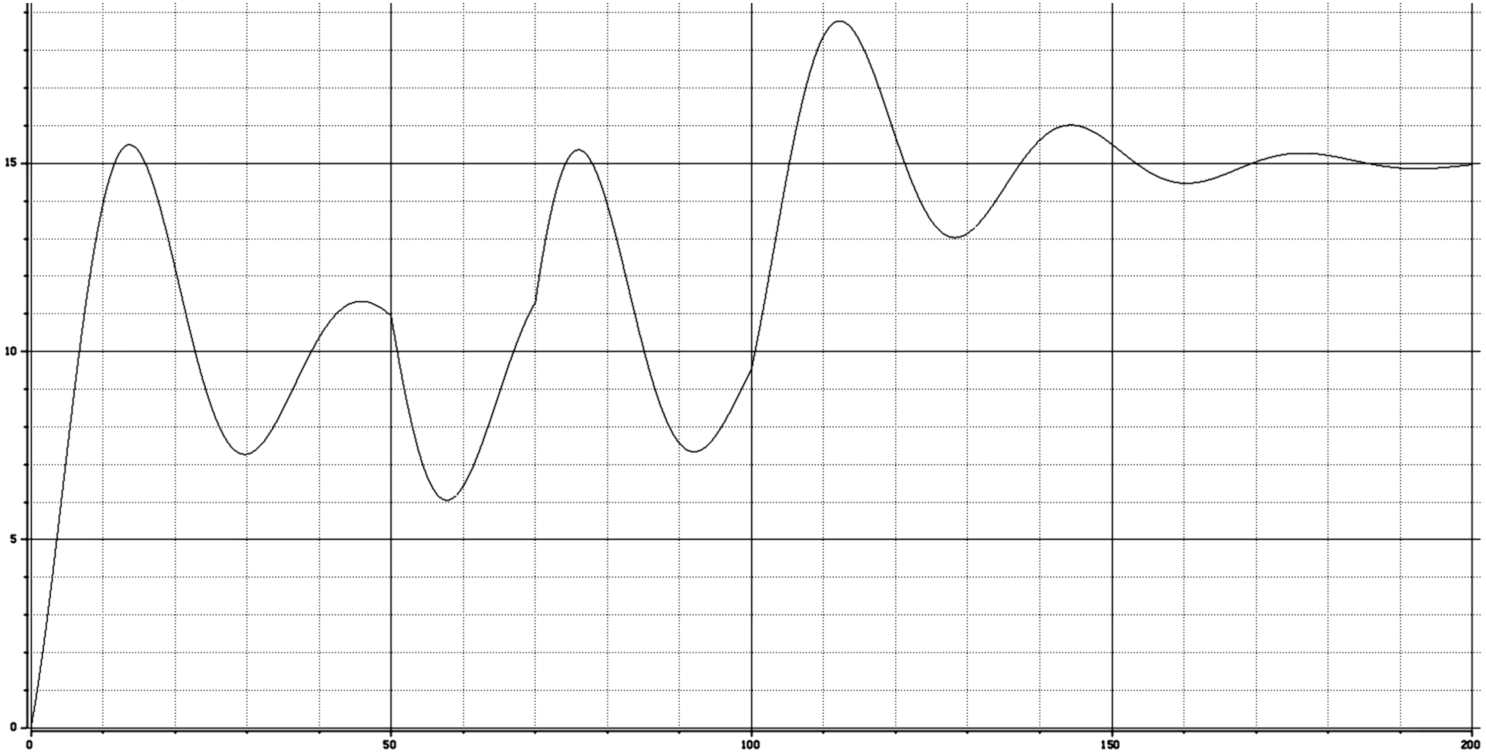
Translator Program

instantiates



Simulation Language Metamodels

Simulation



- Generated simulation files execute the same way on Modelica and Mathworks Tools

Areas for Integration

1. Physical interaction / signal flow simulation.
2. Finite element analysis.
3. Mathematical unification of systems and analysis models.
4. Discrete event analysis and optimization (production, logistics)
5. Sustainability analysis.

Discrete Event Logistics Outline

- What are Discrete Event Logistics Systems (DELS)?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
- Where are we now?
- 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).

<u>Examples include:</u>

- Supply chains
- Manufacturing systems
- Transportation
- Material handling systems
- Storage systems
- Humanitarian logistics
- Healthcare logistics
- Sustainment Logistics
- 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

Discrete Event Logistics Outline

- What are Discrete Event Logistics Systems (DELS)?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
- Where are we now?
- Where do we want to go?

Interest to MBSE Community

- Bring a different domain into the MBSE community
  - In the design of logistics systems, we don't have good SE tools and practices
- Why can MBSE 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

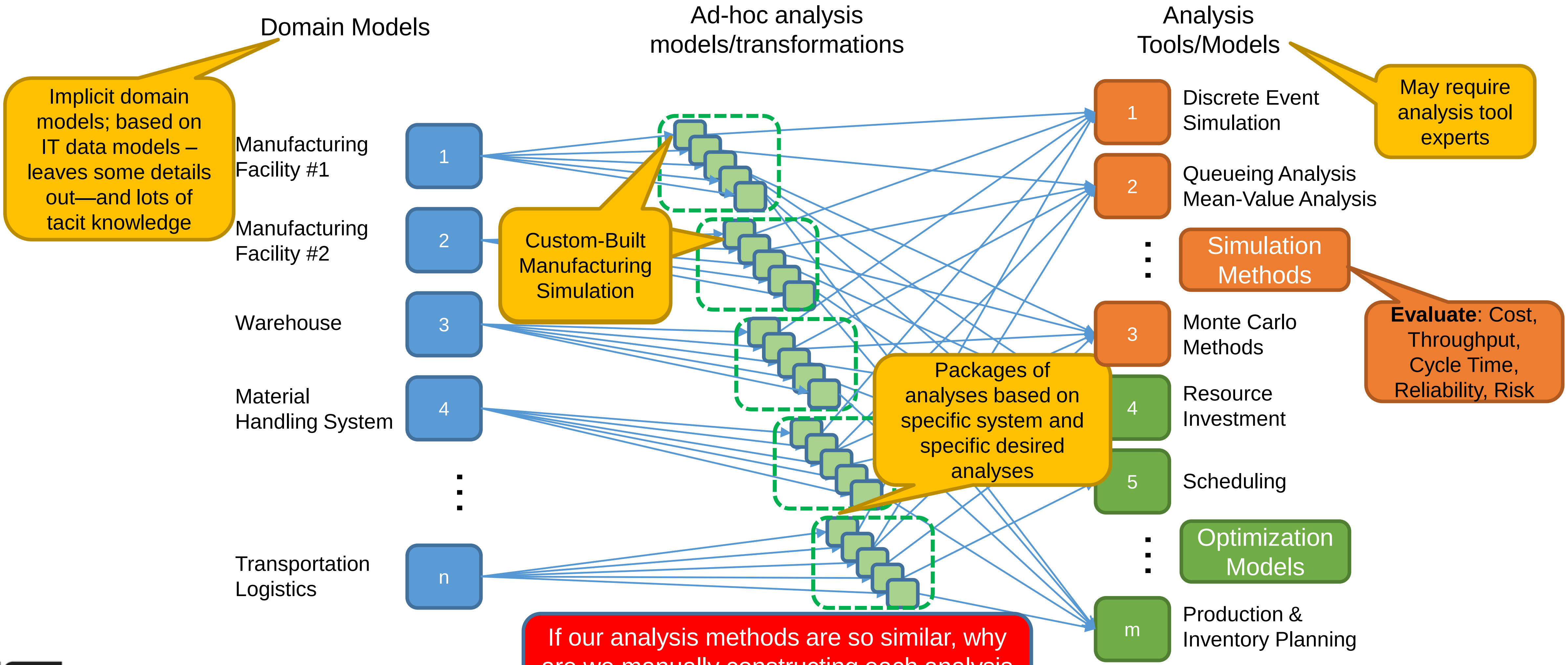
Need for Model-Based Methods

- Current methods and tools are limited for production systems engineering
    - Formal specification & analysis automation
    - Design and teaching

- Documentation & Organization of Knowledge
    - Existing Systems Models (industry)
    - Existing Analysis Models (academia)

- Bridge between system and analysis models
    - Interoperability between different analysis models of the same system
    - Greater reusability of analysis: collaboration and automation
    - Modeling & Simulation Interoperability (MSI)

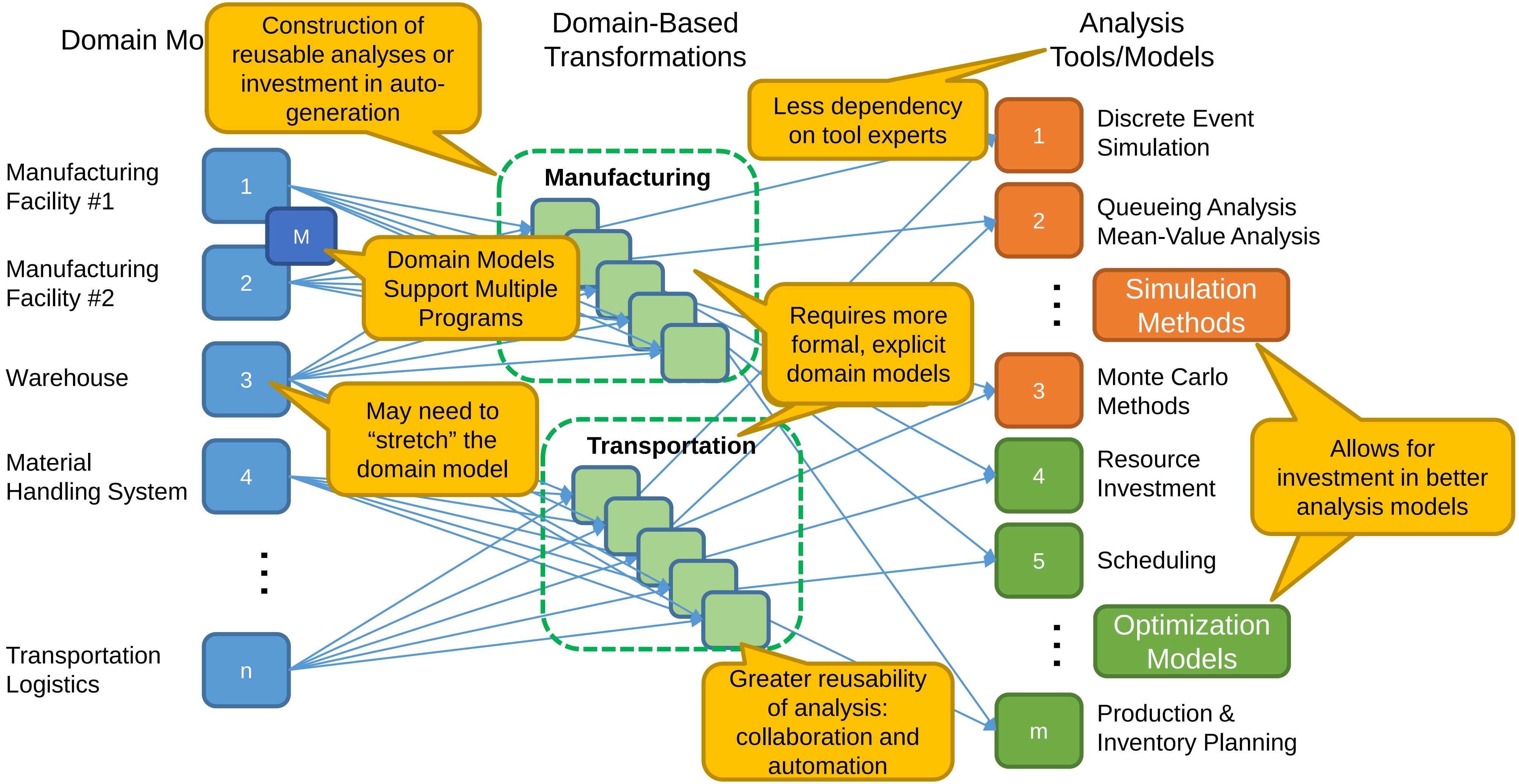
System Model to Analysis Model Transformation: Status Quo – Manual Ad-Hoc Analysis Generation



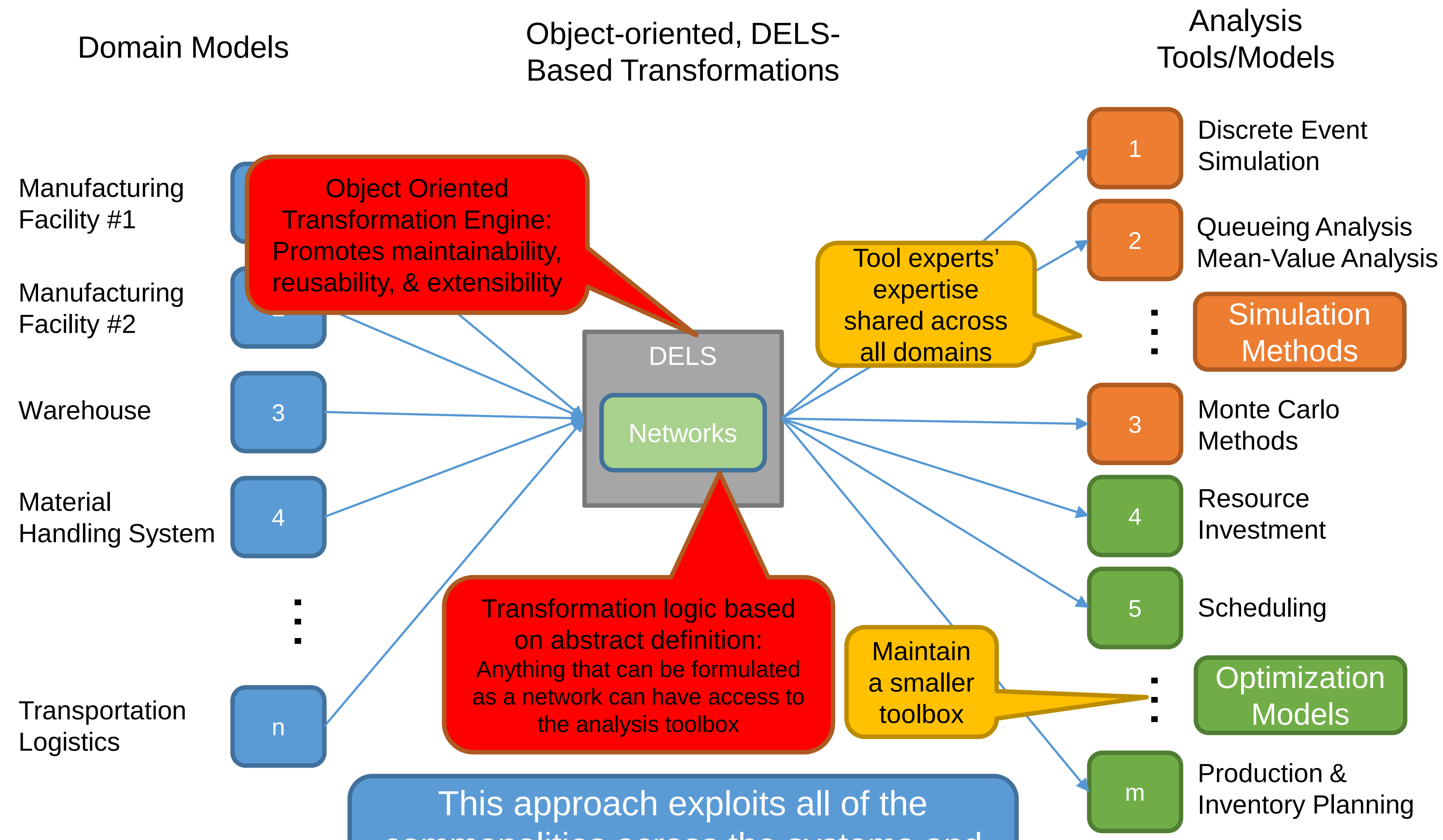
If our analysis methods are so similar, why are we manually constructing each analysis model for each system?

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

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

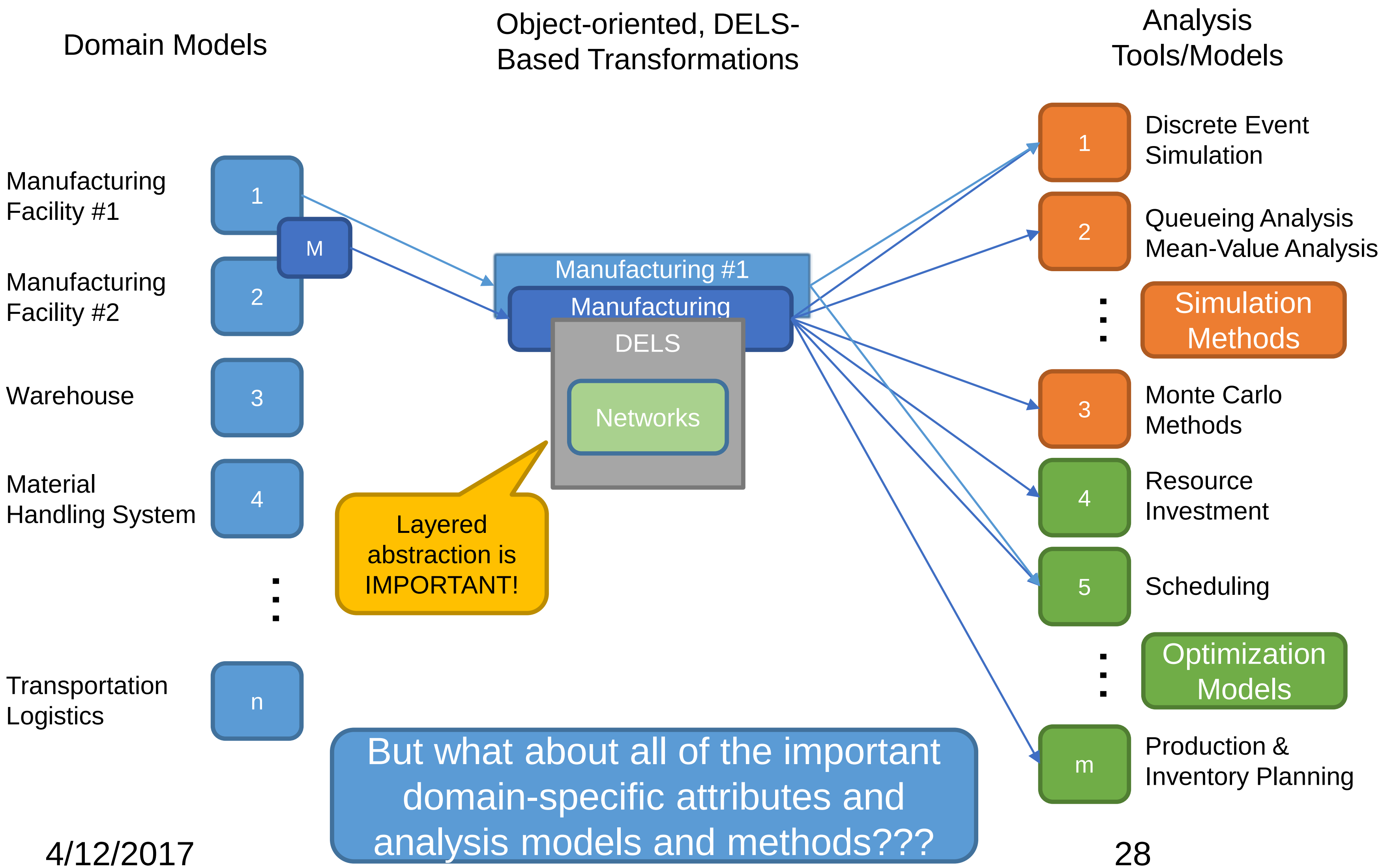


System Model to Analysis Model Transformation: M2M Methods Based on DELS Abstraction



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



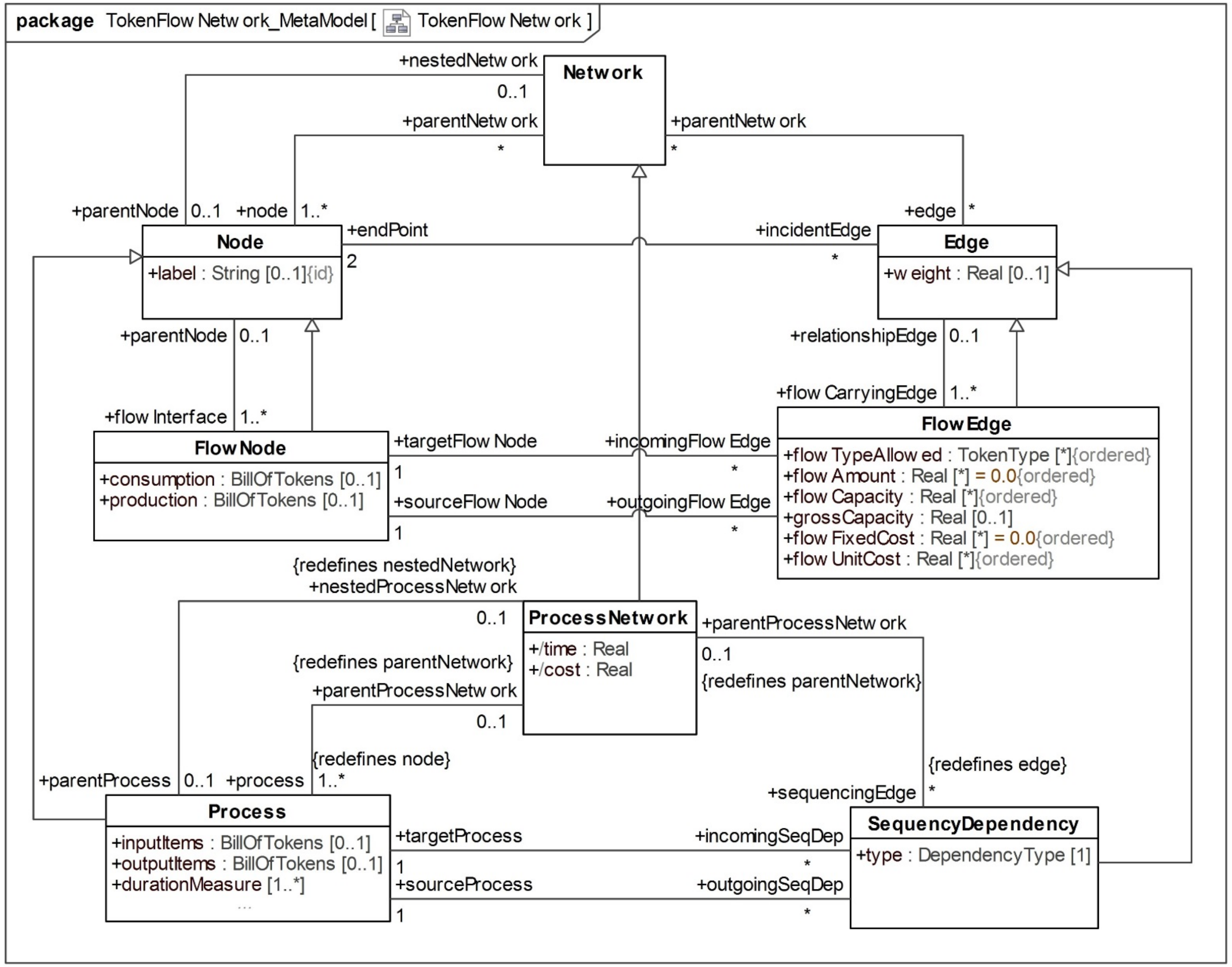
Discrete Event Logistics Outline

- What are Discrete Event Logistics Systems (DELS)?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
- Where are we now?
- 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.

Network Abstraction

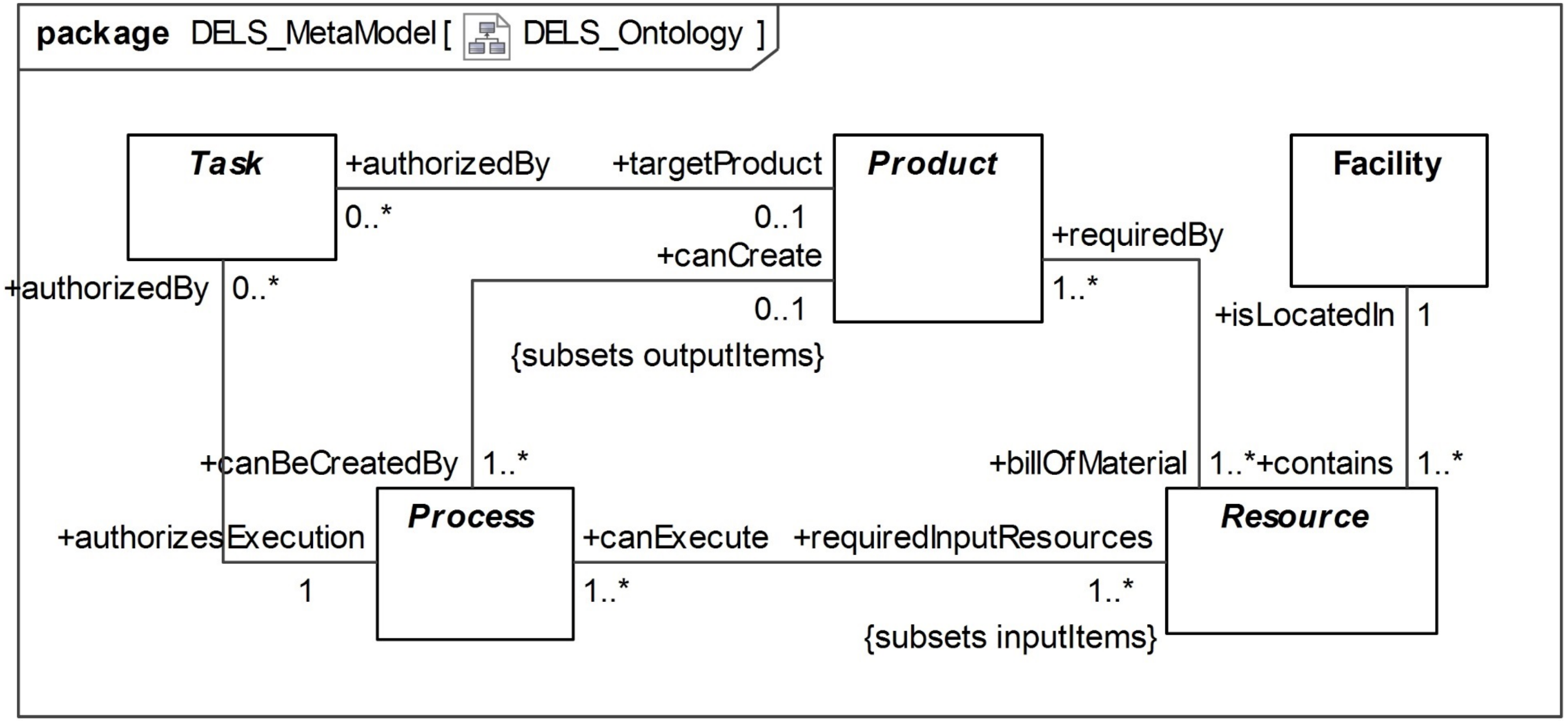


Networks, Flow Networks, and Process & Queueing Networks

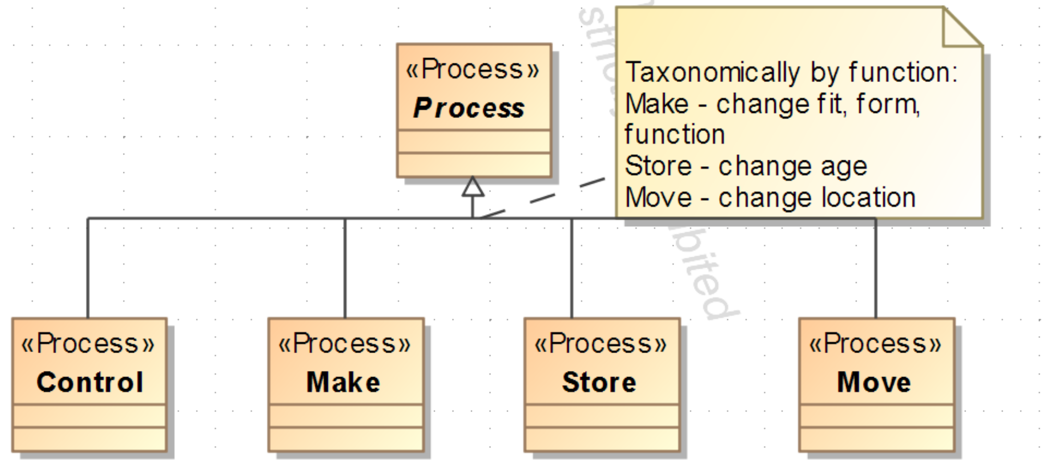
- Form the basis of many analysis methods in the industrial engineering and operations research (IEOR) domain.
- Abstract and reusable across many related domains

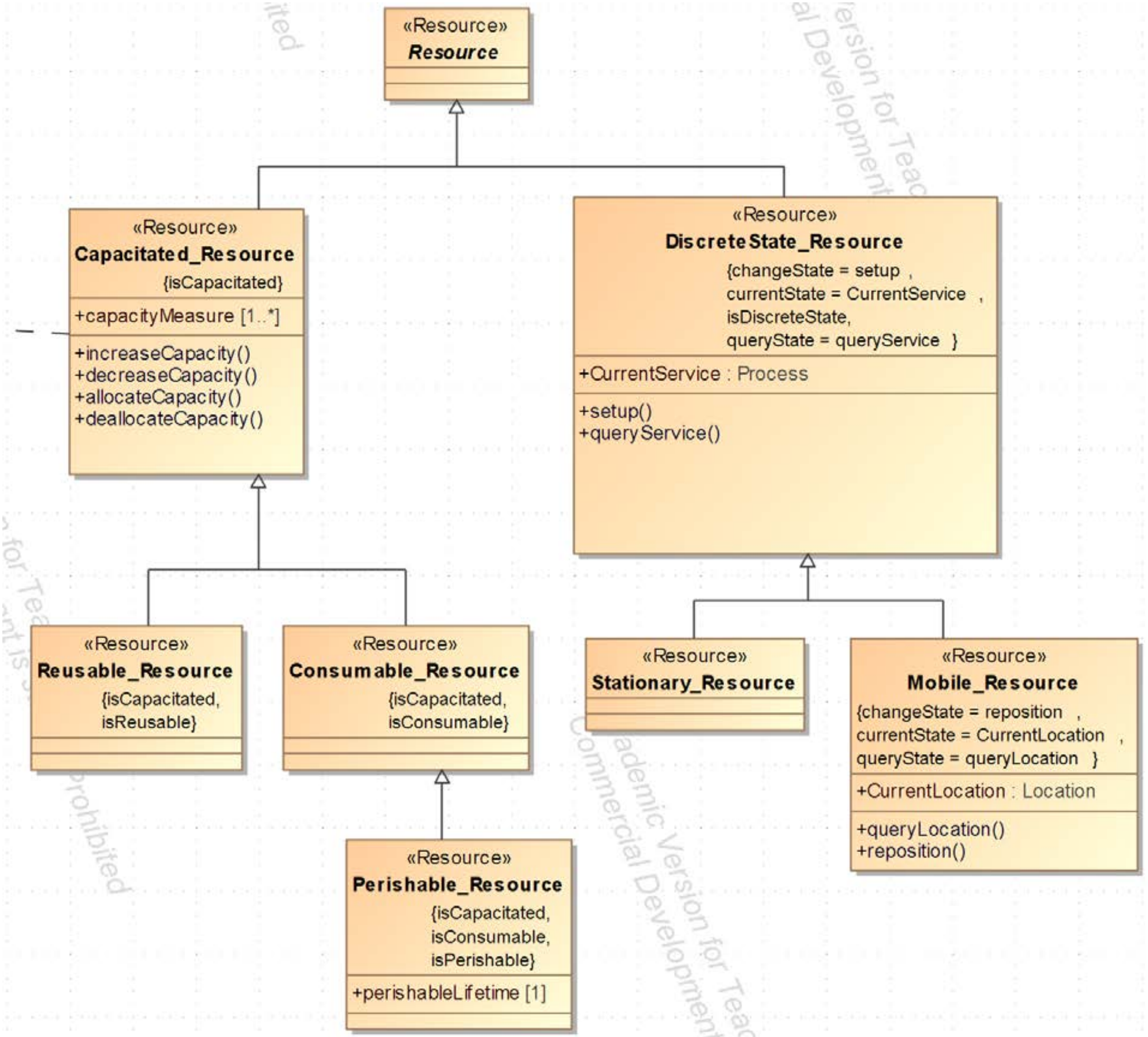
DELS Behavior – Product, Process, Resource, Facility

Fundamental concepts necessary to describe the behaviors of which the DELS is <u>capable</u>.



Taxonomies of DELS Behavior

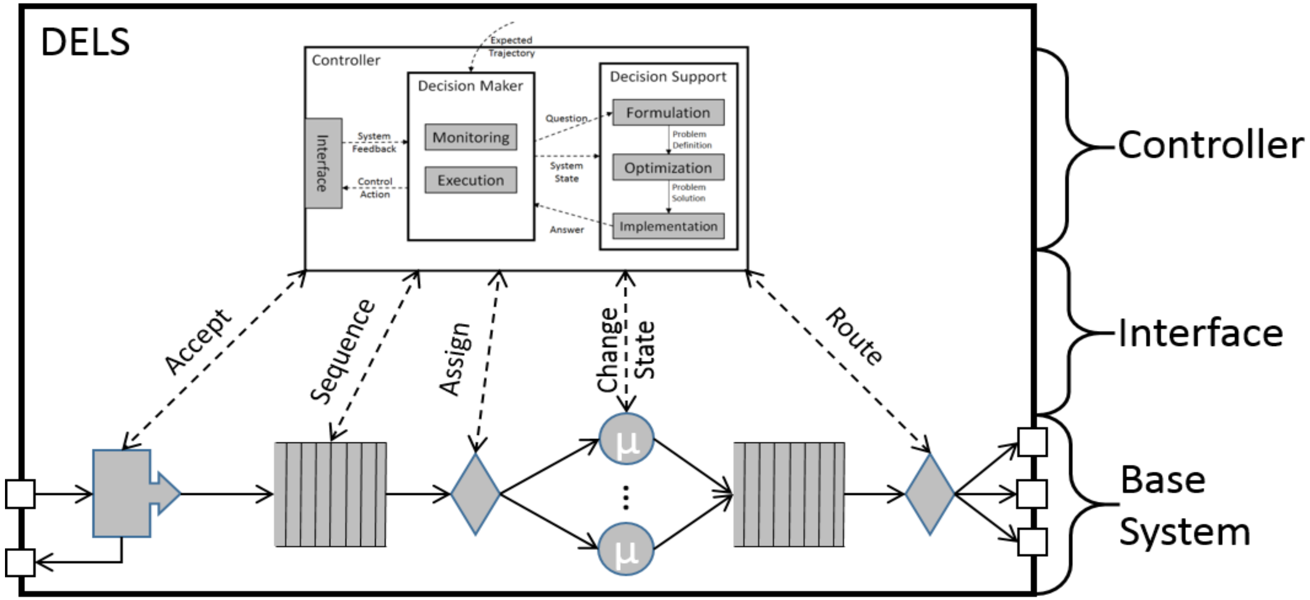




Can be elaborated to support more expressive and fine-grained system models, capturing more particular aspects of classes of systems.

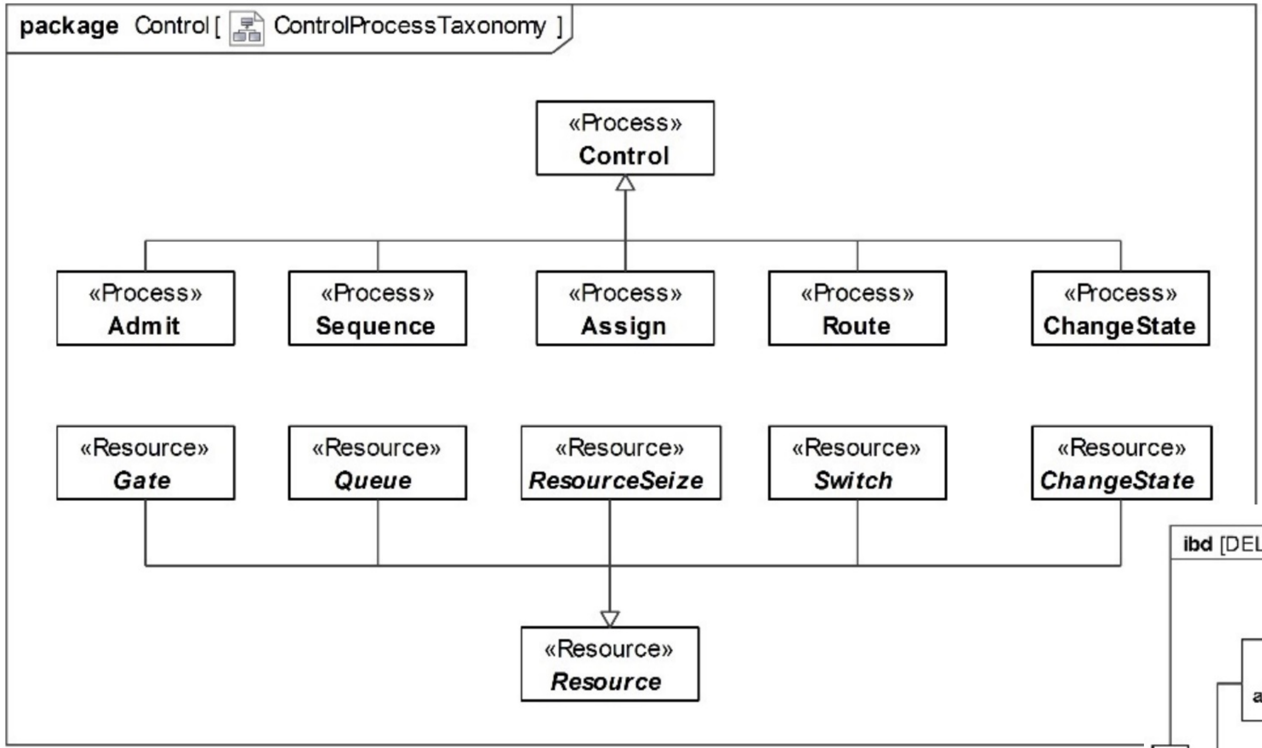
Operational Control

Manipulating flows of tasks and resources through a system.



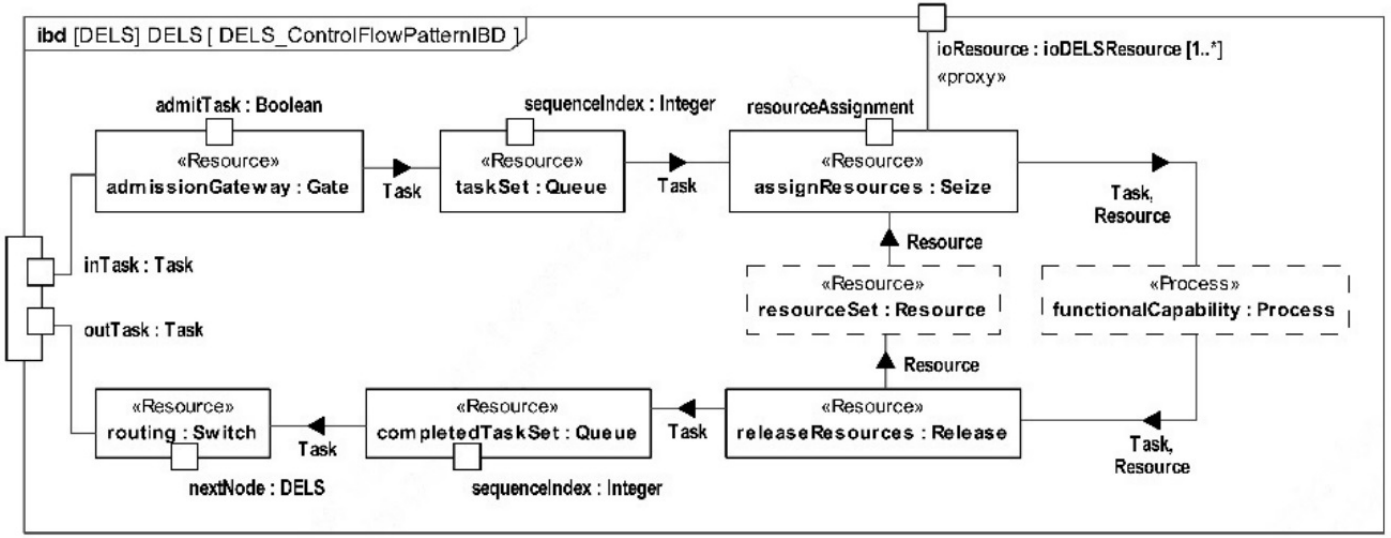
- Which tasks get serviced? (Admission/Induction)
- When {sequence, time} does a task get serviced? (Sequencing/Scheduling)
- Which resource services a task? (Assignment/Scheduling)
- Where does a task go after service? (Routing)
- What is the state of a resource? (task/services can it service/provide)

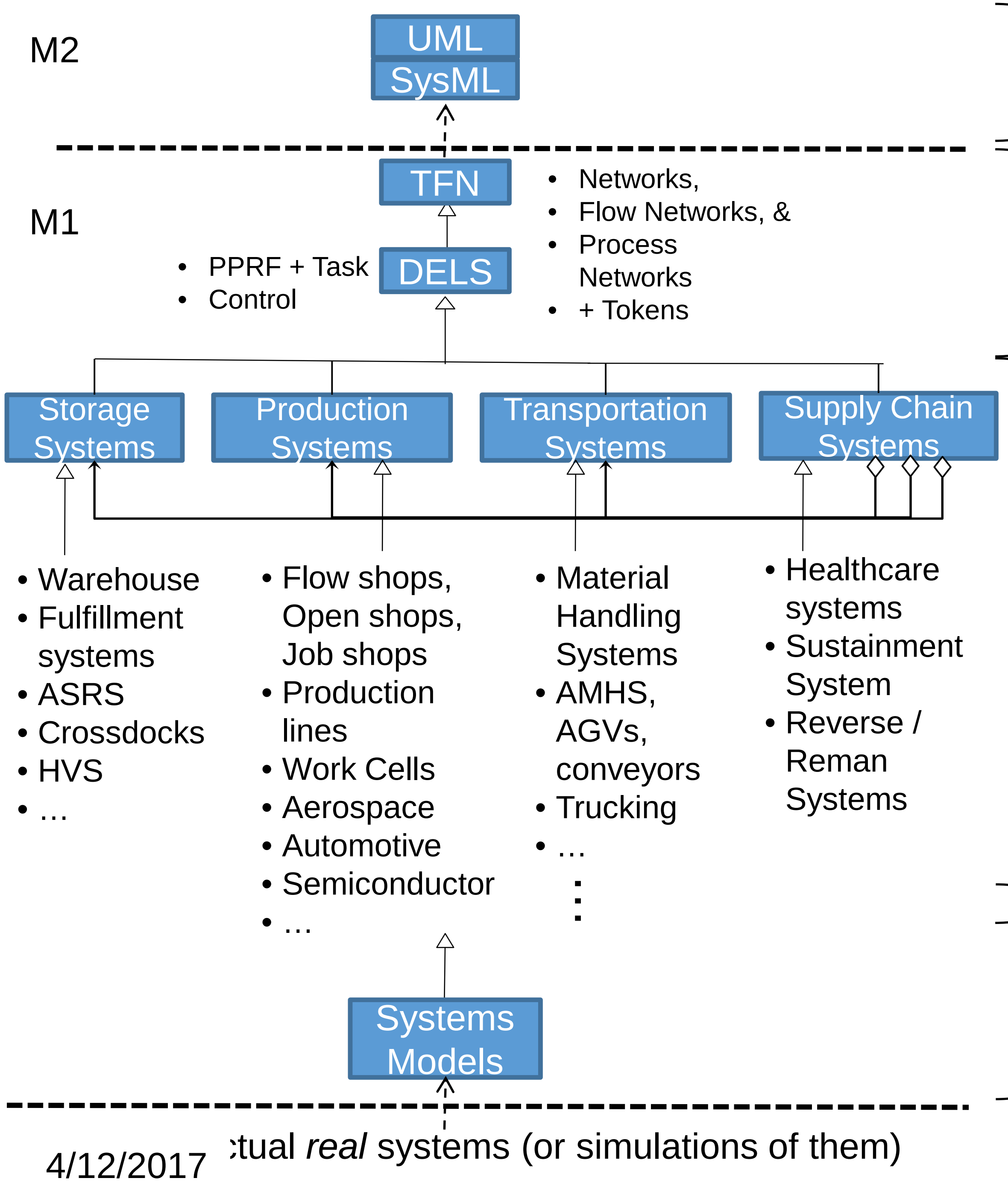
Operational Control



Extends the PPRF definition to special classes of control processes and resources

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





Language Layer

- May also include a TFN & DELS DSL

Top of M1

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

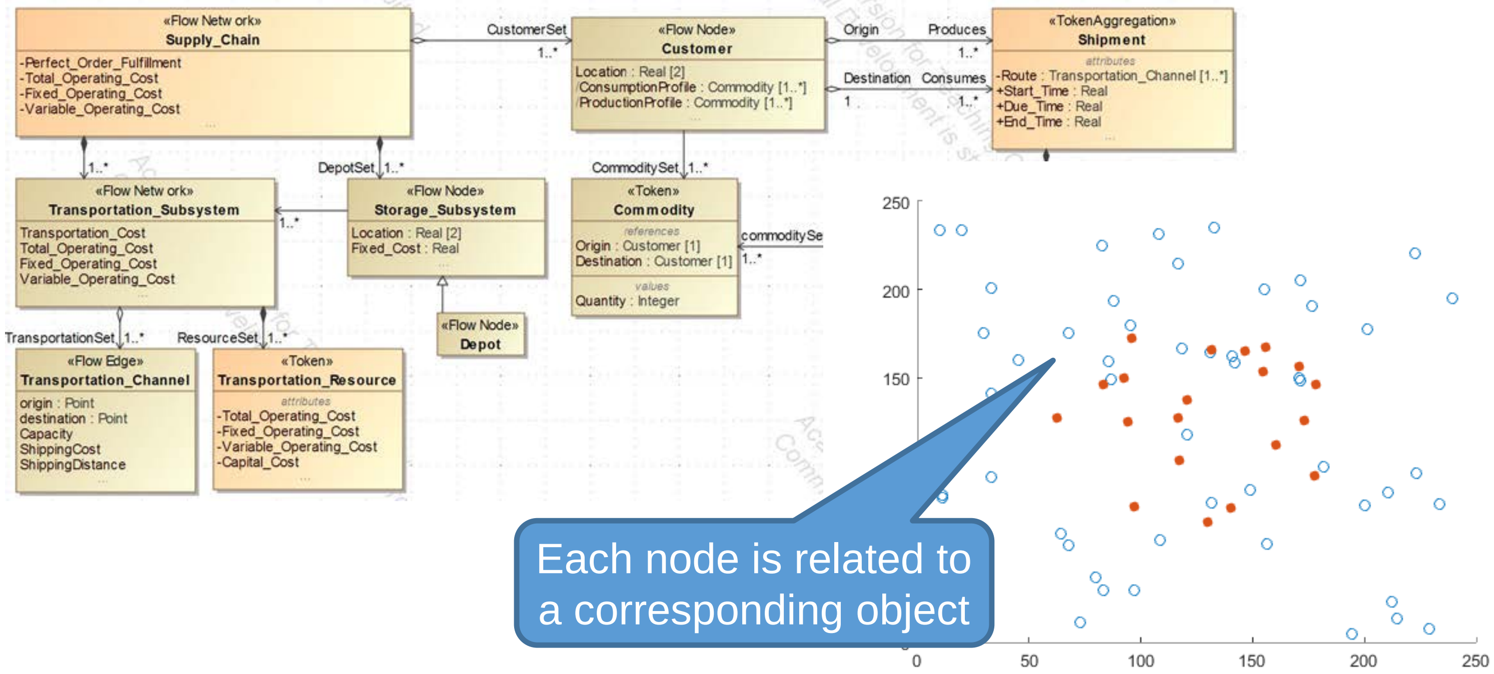
Middle of M1

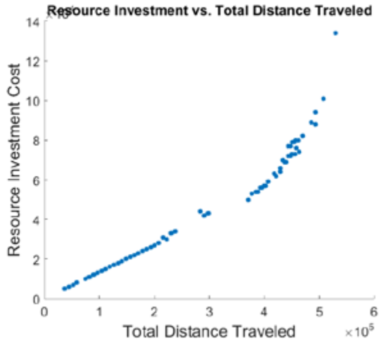
- **(sub-) Domain-specific reference models and architectures**
- Generalization Set aligns with STORE, MAKE, & MOVE processes

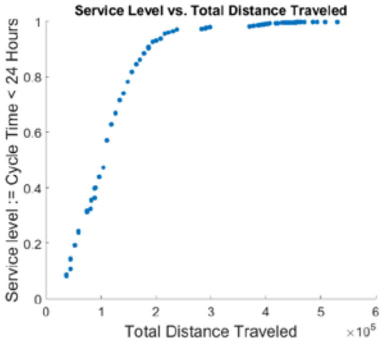
Bottom of M1

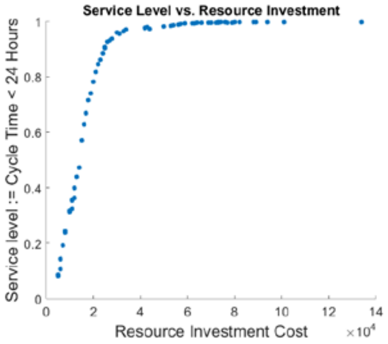
- **System Models**
- "as-built" or "specification" models

System-Analysis Integration – Use Case









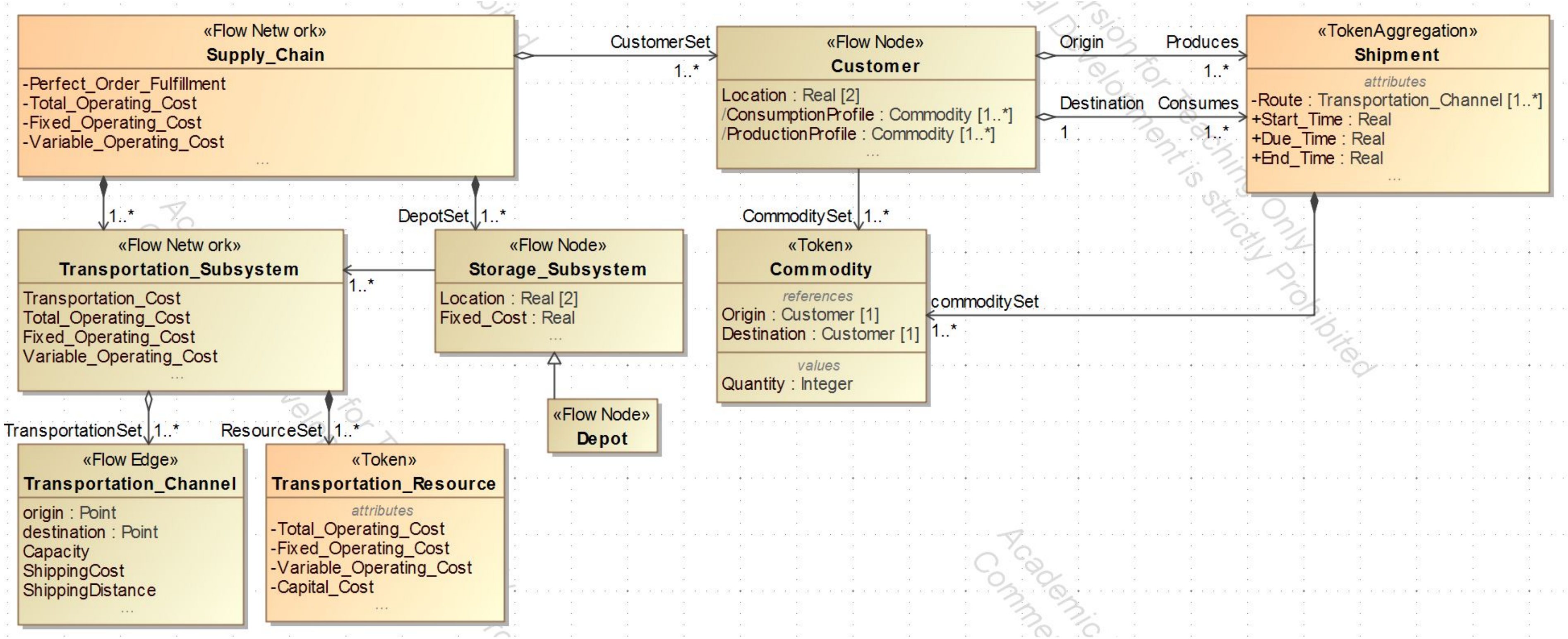
Strategy:

- 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 reference model provides a pattern for constructing conforming system instance models and analysis models.

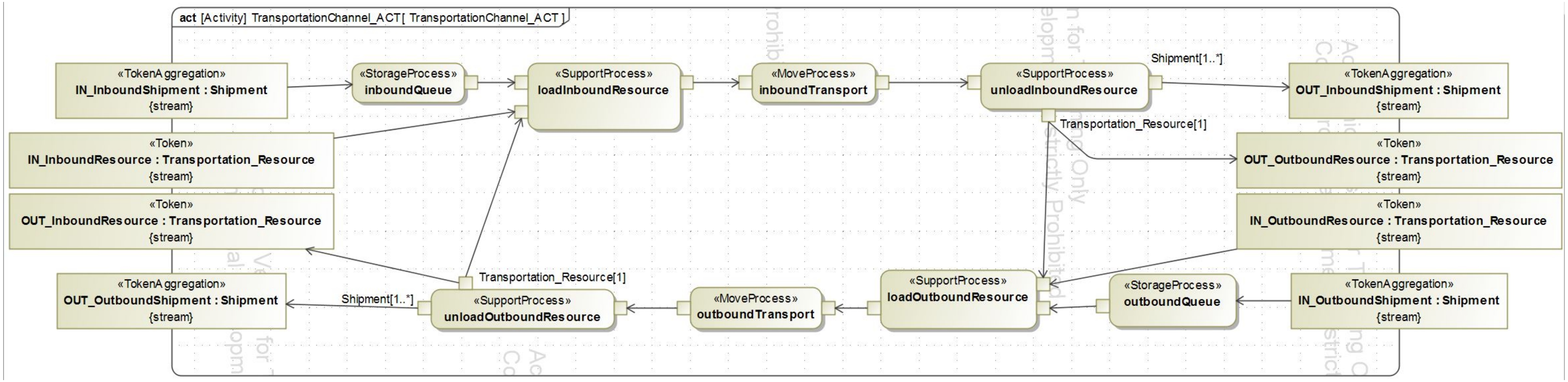
- ➢ The system of interest is a distribution supply chain.



Transportation Channel Behavior

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

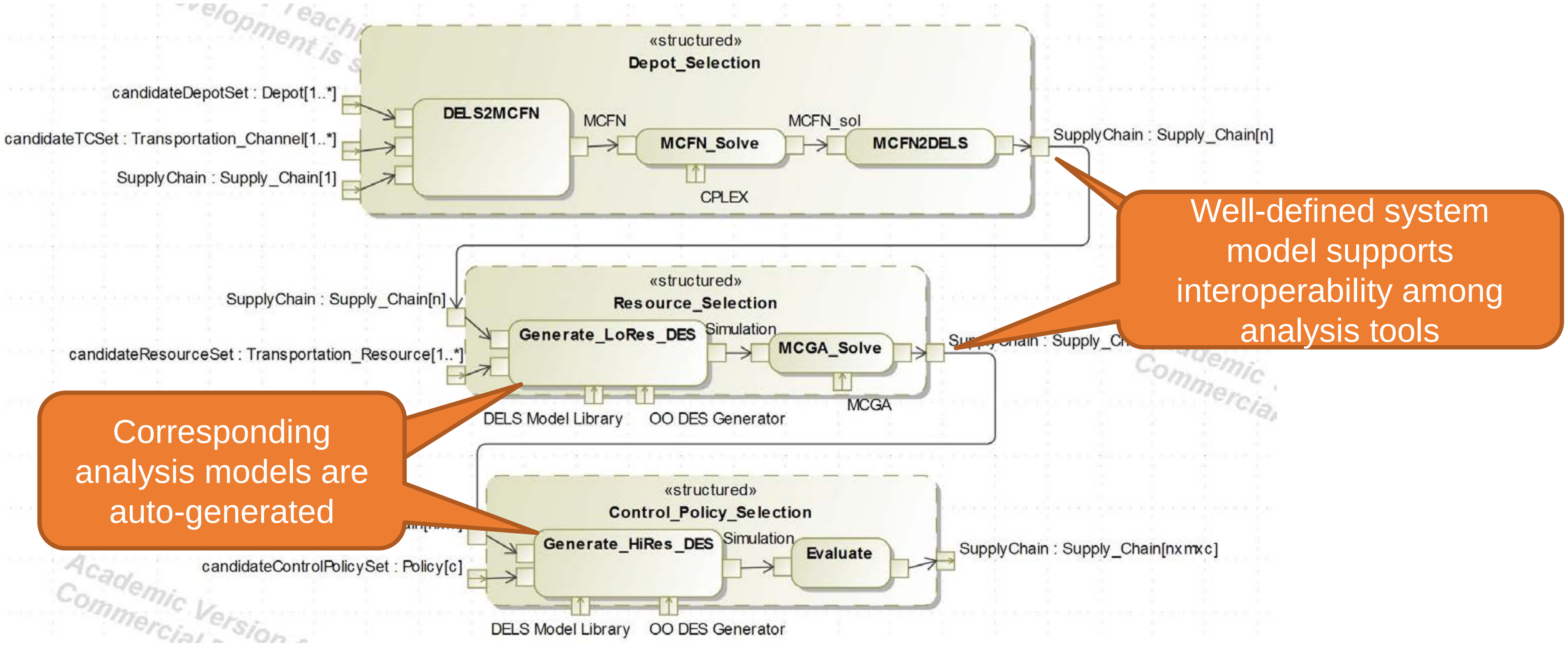
- Component-based generative methods for simulation models
- V&V of model library components, compose models from components



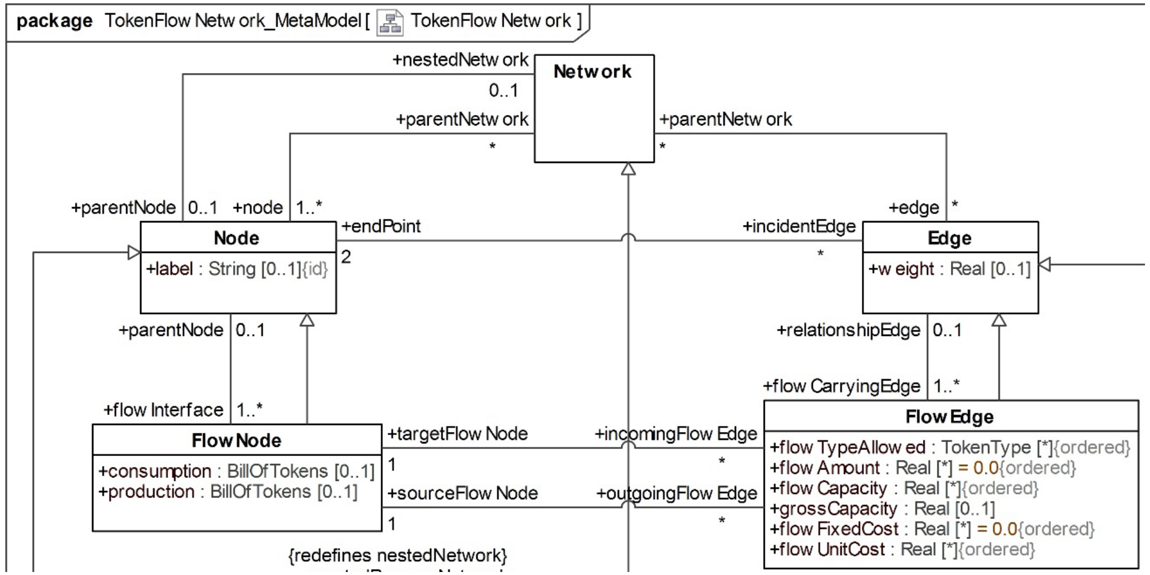
Analysis Methodology Overview

Hierarchical design methodology 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

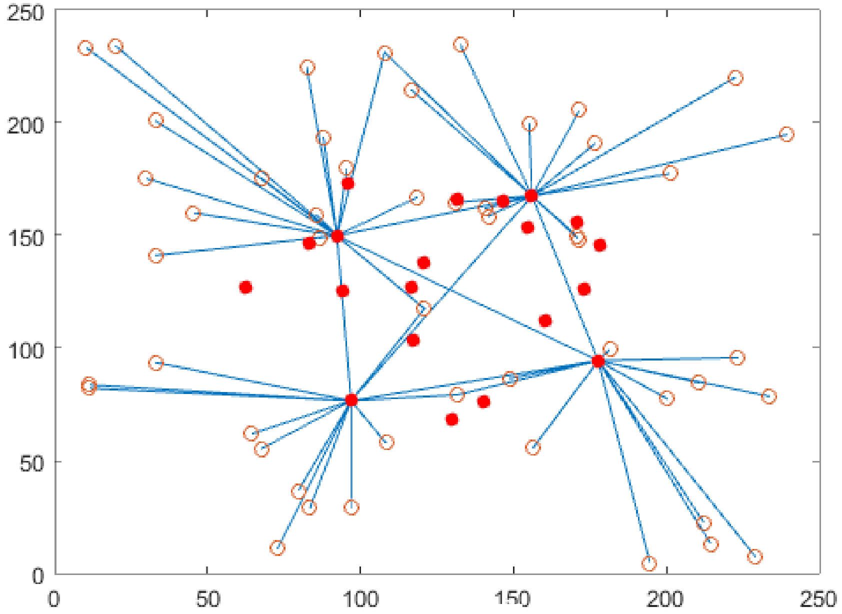


Optimize Network Structure – Where to put the depots?



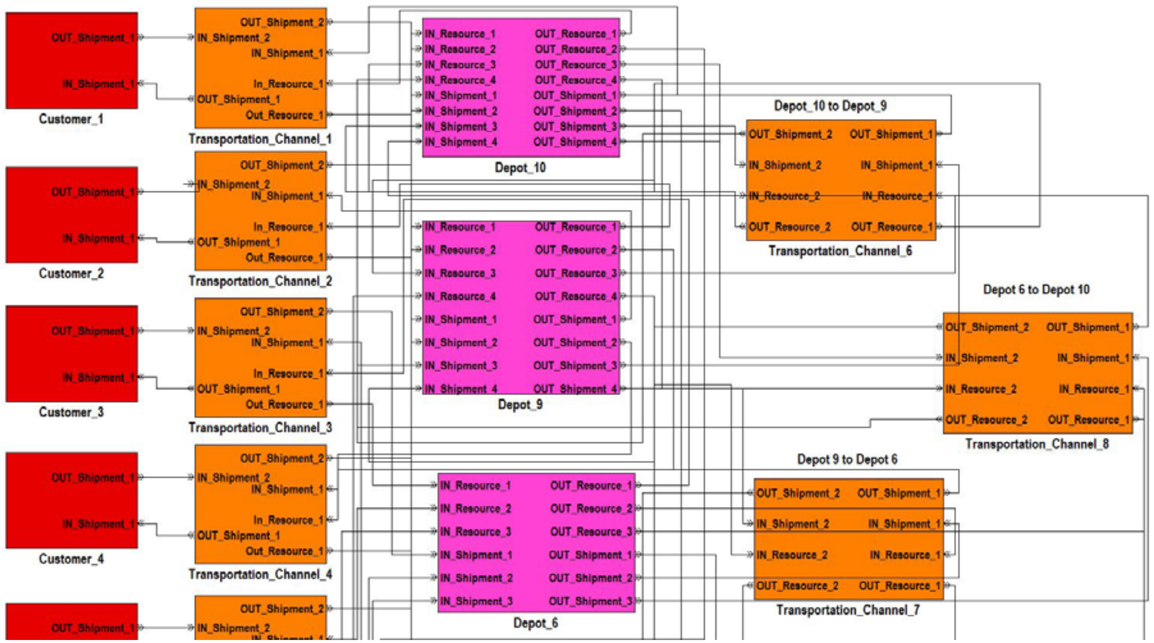
Goal: Reduce the computational requirements of optimizing the distribution network structure.

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



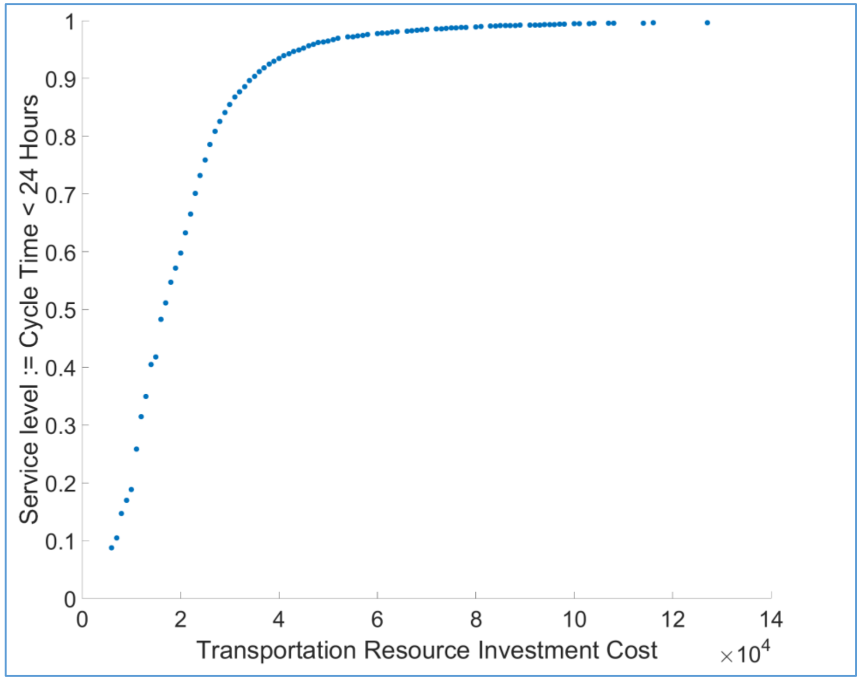
- Abstract the Supply Chain model to a Flow Network model that forms the backbone of the analysis model
  - Aggregate and approximate the flows and costs
- Solve MCFN using a COTS solver (CPLEX)

Resource Selection – How many trucks?



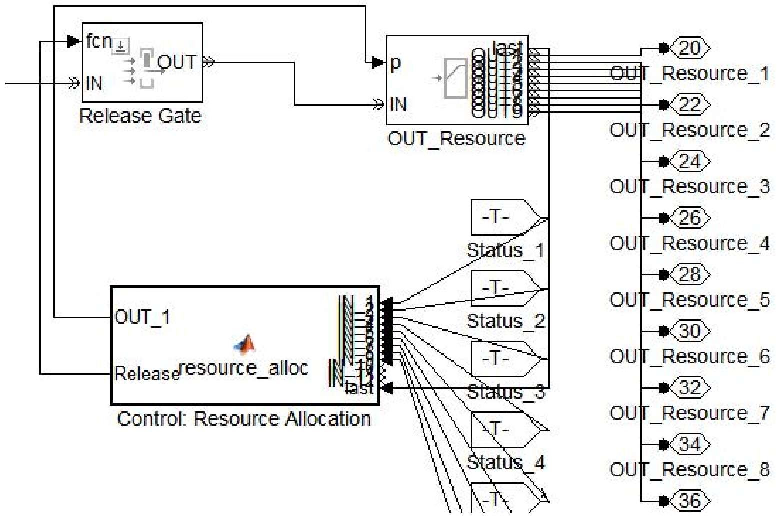
Goal: Capture and evaluate the behavioral aspects of the system using discrete event simulation.

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

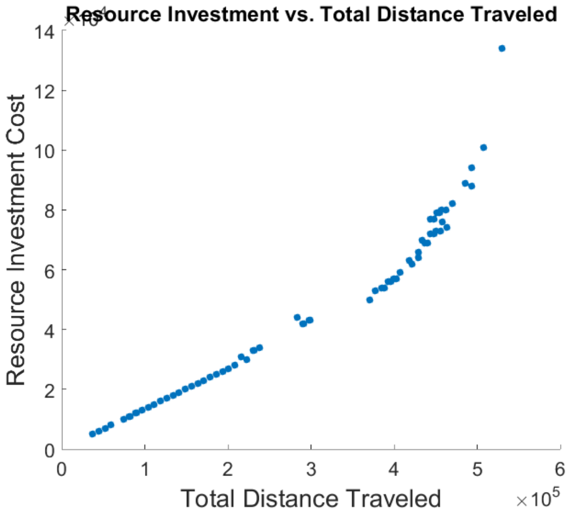
Configure Control Policies – Which Truck? When?

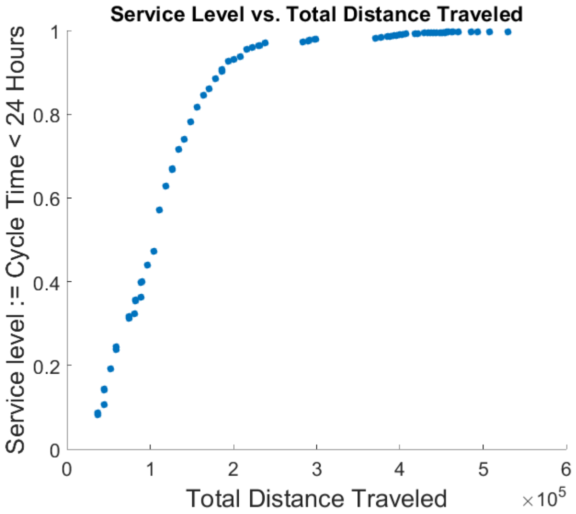


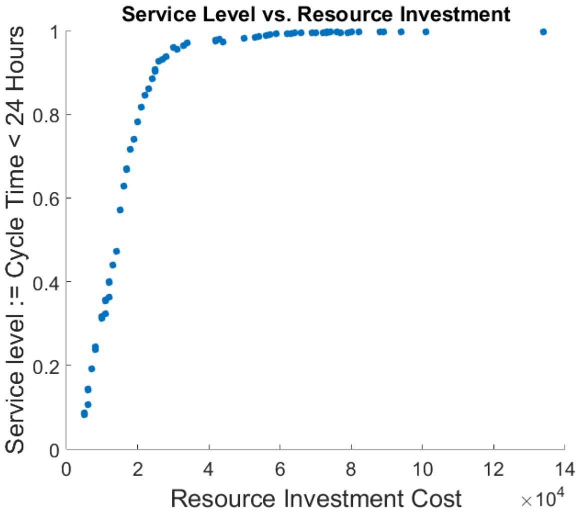
Goal: Select and design a detailed specification of the control policies for assigning trucks to pickup/dropoff tasks at customers.

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

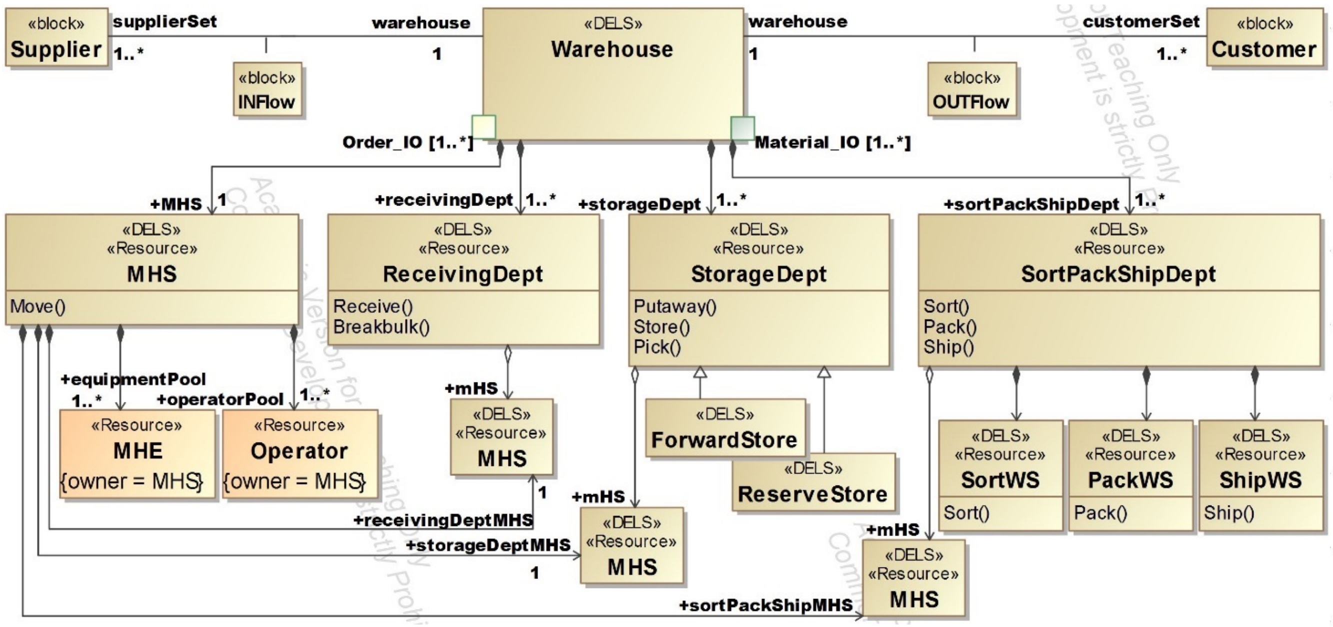
Trade-off Service Level, Capital Costs, and Travel Distance

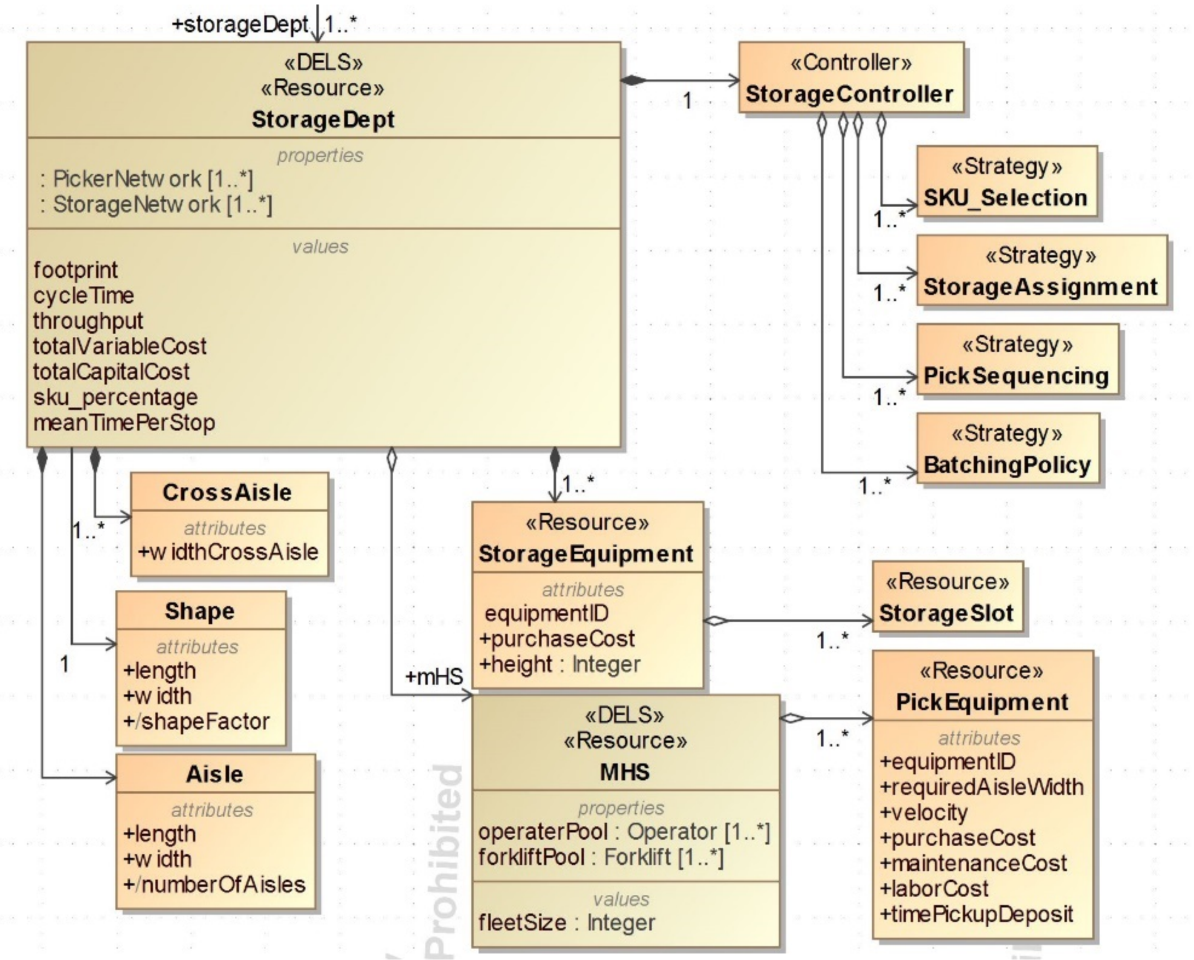






Warehouses

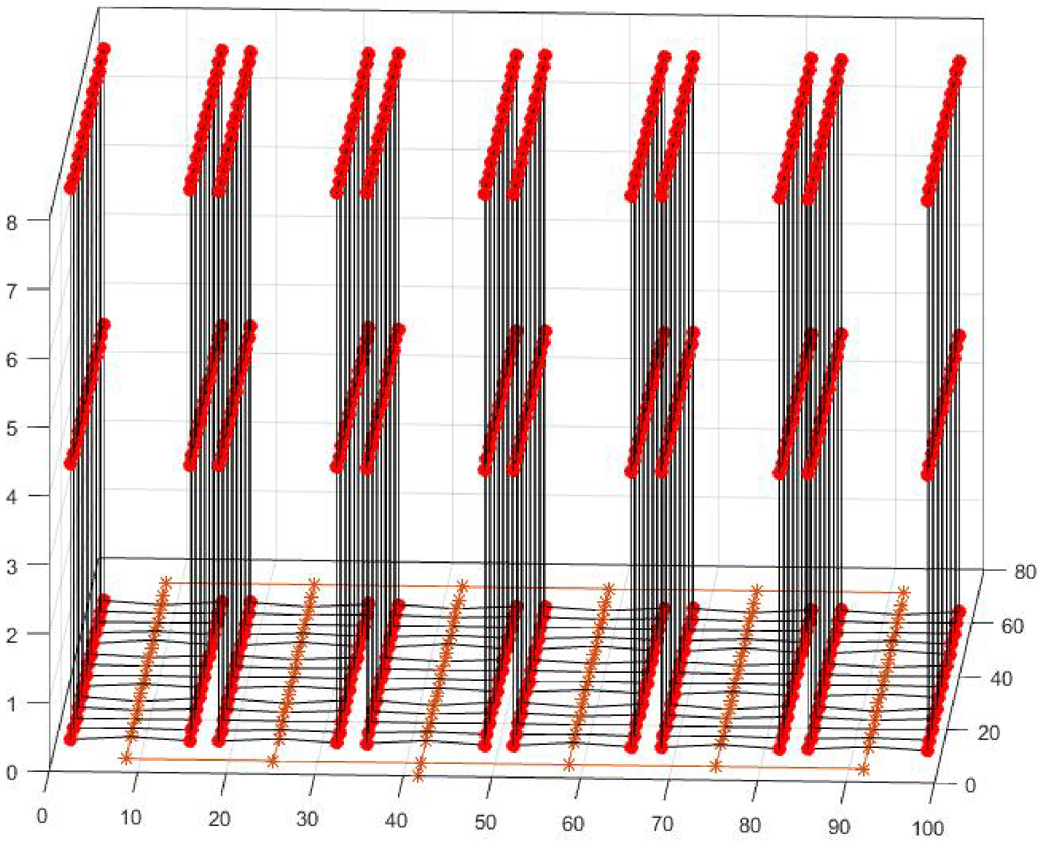




<u>Same Strategy:</u>

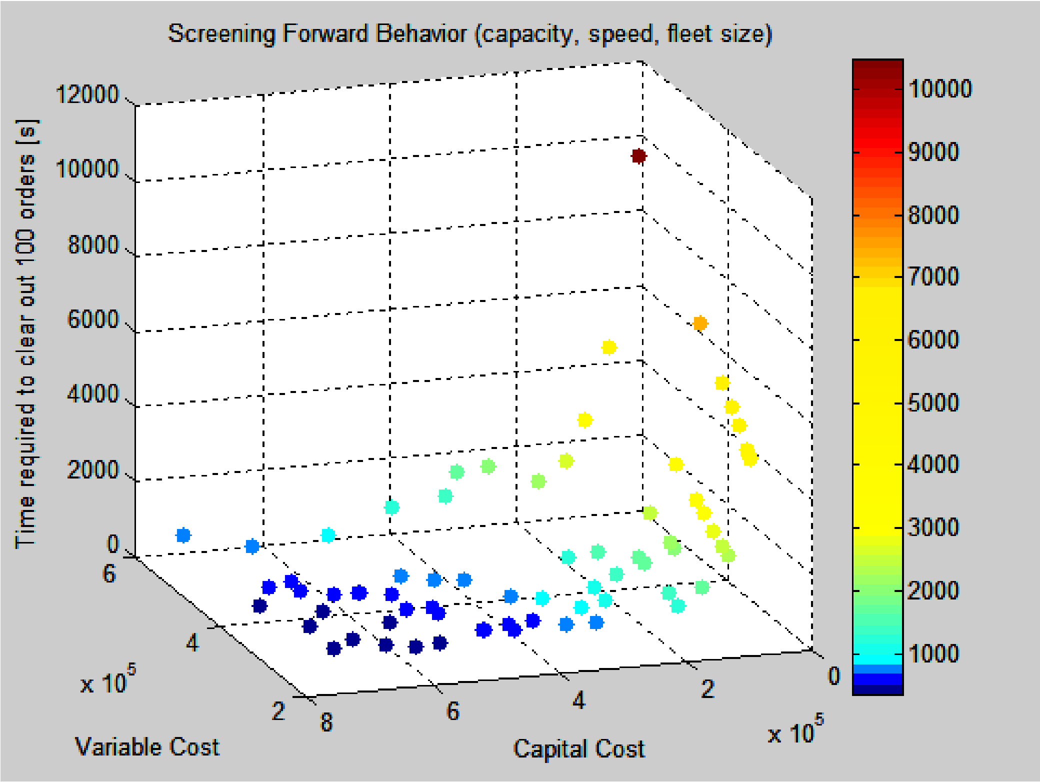
- Start with a system model,
- Generate simulation models and analysis models (decision support),
- Generate candidate designs.

Analysis Model Generation



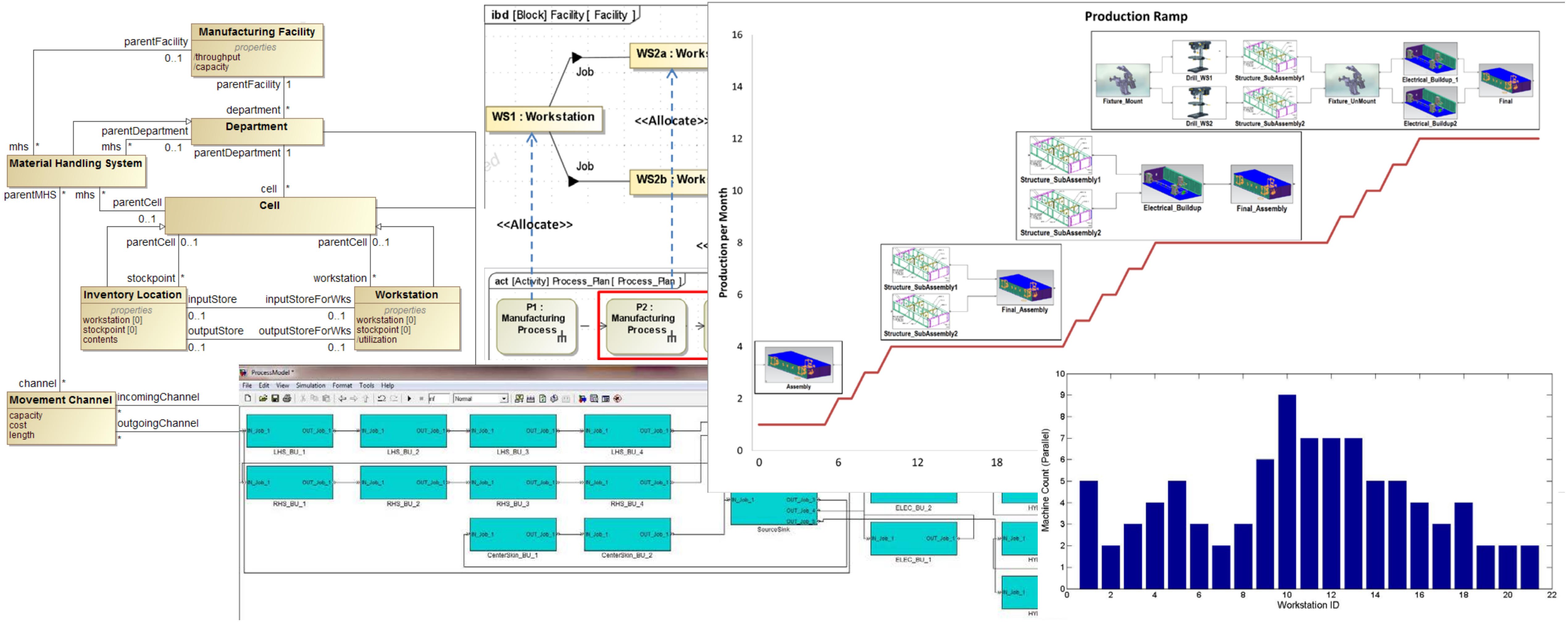
Metrics to support decision making:

- 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



Why do it this way?

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

Discrete Event Logistics Outline

- What are Discrete Event Logistics Systems (DELS)?
- What are the fundamental challenges for DELS?
- Why do we need system models and MBSE?
- Where are we now?
- Where do we want to go?

Where do we want to go?

- INCOSE MBSE Initiative Challenge Team on DELS Modeling
  - Single community for modeling DELS

- 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

SAI Overview

- Project organization
- Motivation and approach
- Areas for integration
- Summary

SAI Summary

- Multi-year federal project on integrating systems modeling and analysis.
- Topic areas so far
  - Physical interaction and signal flow simulation
  - Finite element analysis.
  - Mathematical unification of systems and analysis models.
  - Discrete event analysis and optimization (production, logistics).
  - Sustainability analysis.
- Results will be publicly available.

More Information

- SAI Project description
  - http://www.nist.gov/el/msid/syseng/smsi.cfm
- Project lead
  - Conrad Bock, conrad.bock@nist.gov
- Discrete event logistics lead
  - Tim Sprock, timothy.sprock@nist.gov