FUTURE OF MH RESEARCH: ONE VIEW

LEON MCGINNIS
PROFESSOR EMERITUS
GEORGIA TECH

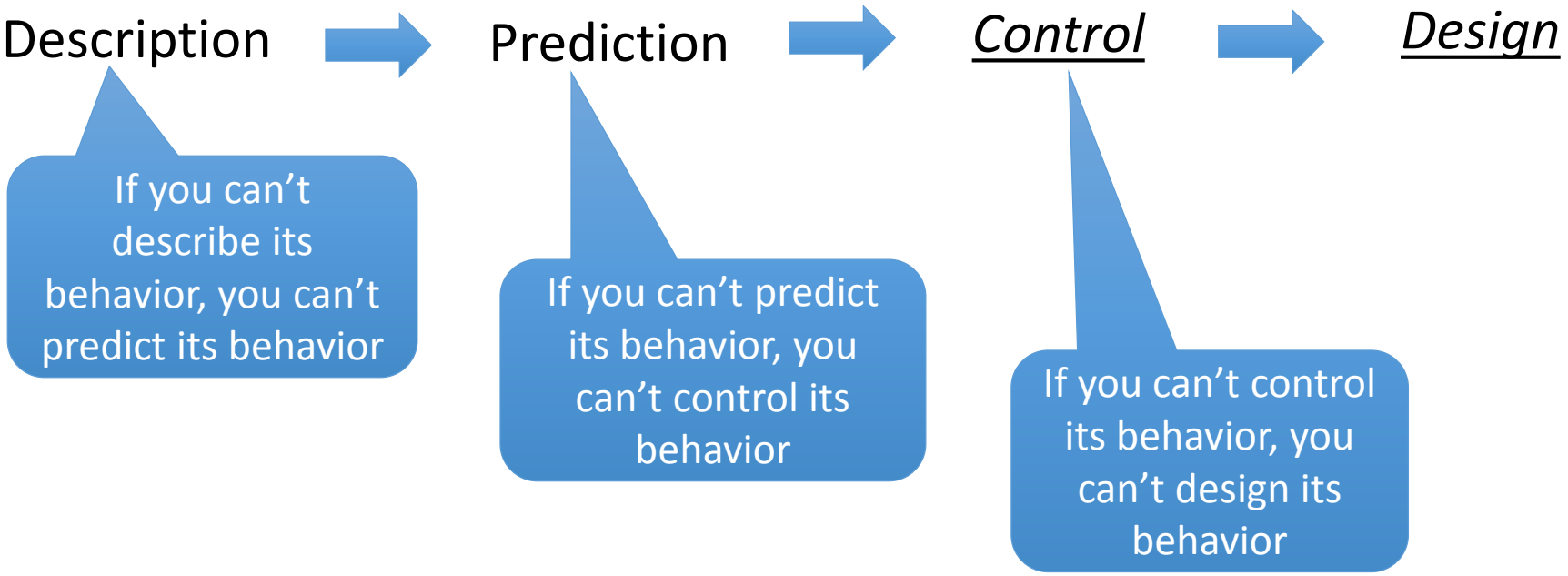
CREATING THE NEXT®


BANG


Keyword

Georgia Tech

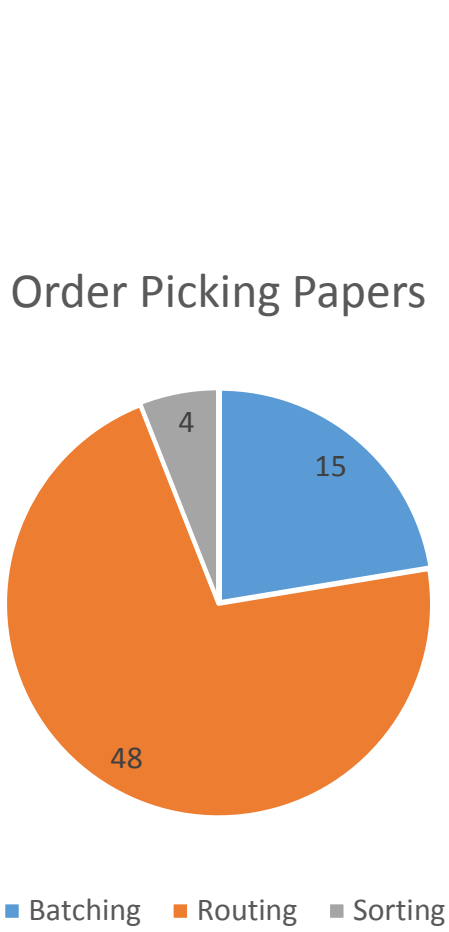
Engineered Systems Design

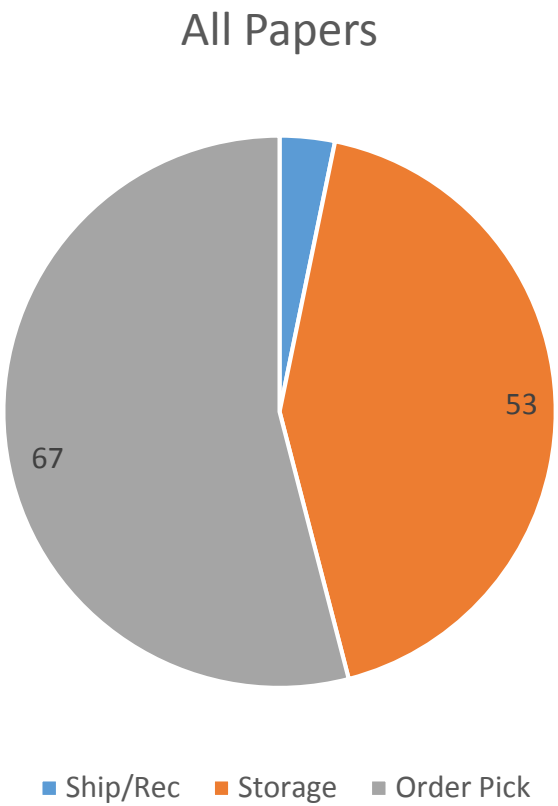


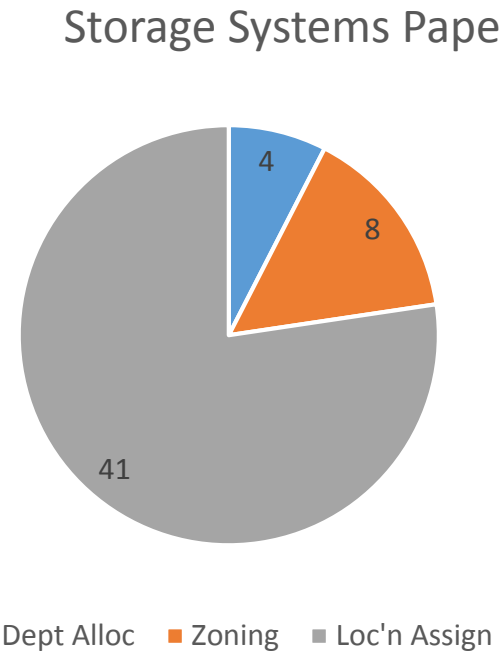
All our research is predicated on our ability to *describe* the engineered system of interest.

It is also *limited* by that ability!

How do we do our research?







From: European Journal of Operational Research Volume 177, Issue 1, 16 February 2007, Pages 1-21

Inescapable Conclusion

The collected body of research on warehousing is very highly concentrated on issues where the system of interest can be "adequately" described using our standard analysis methodologies—statistics, optimization, queuing theory, or discrete event simulation.

And some really great research has been done that has amazing impacts on the practice of MH!

Our “system modeling language” is, in fact, our analysis modeling language.

Like every generalization, this one is not 100% true!

These languages may be really good at what they are good at, but …

- limited fidelity
- limited scope
- limited “consumability”

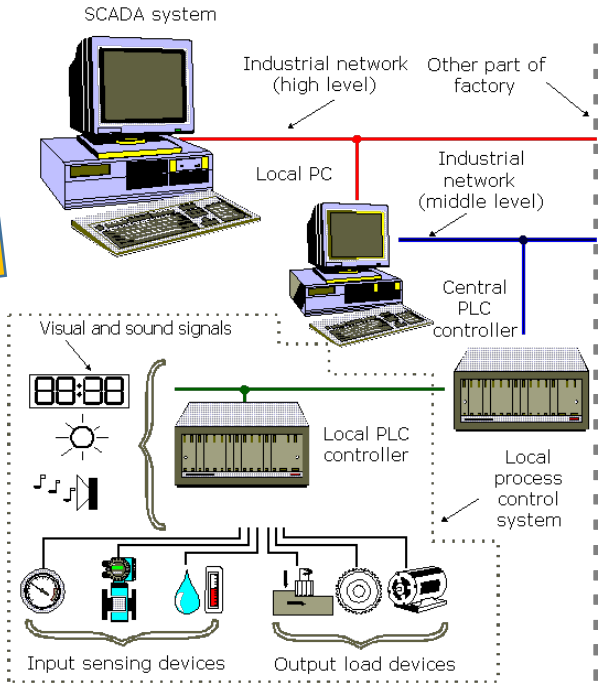
Consequences:

1. We are forced to make assumptions that greatly simplify the real problem, and we rarely are able to test the results to understand approximation error.
   - Steady state, expected value results are “good enough”
2. We turn away from problems for which an adequate system description cannot be created as one of these standard analysis approaches.
   - High density storage/mixing systems
   - Highly capacity constrained sorting systems
3. Our research does not achieve its potential for impact on practice

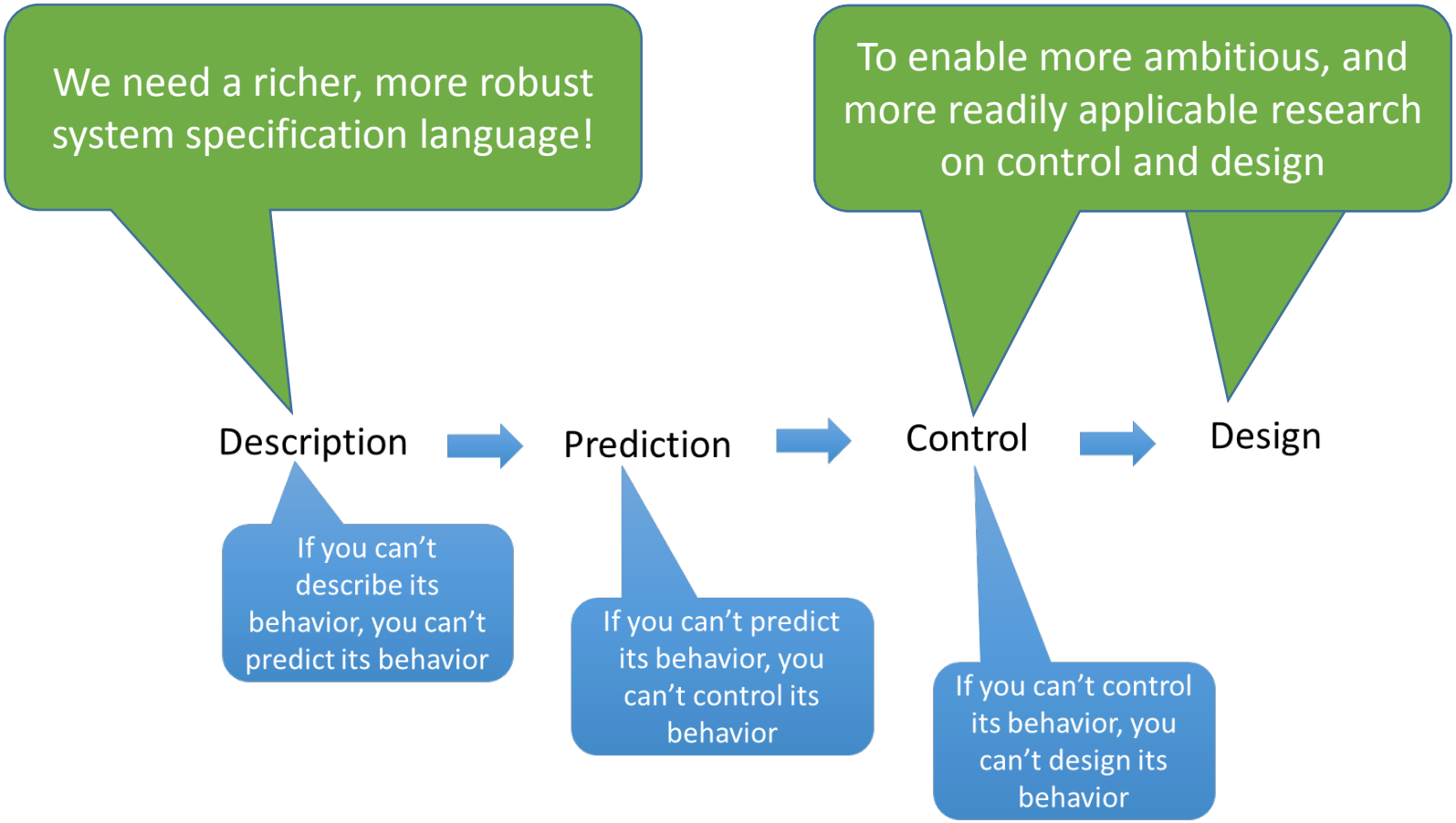
Worst of all:

Results, recommendations from our analyses

Implementation Gap



How to expand our horizons?



A pattern to learn from

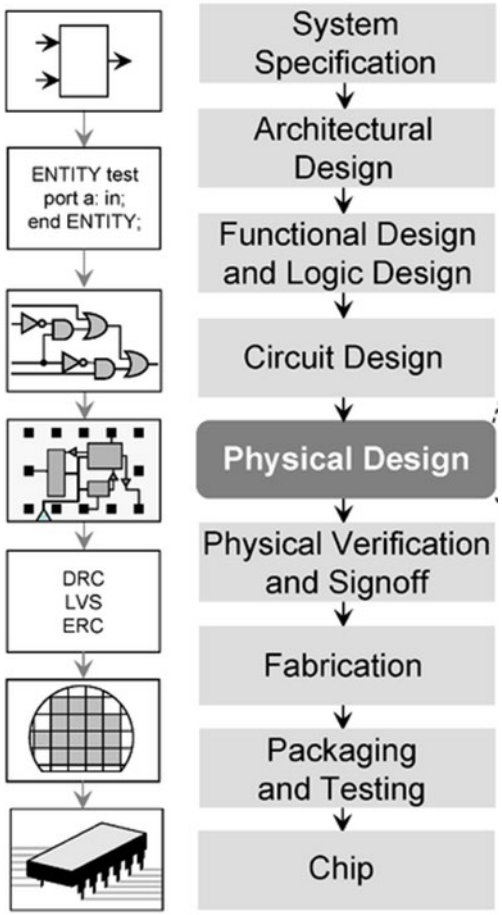
VHDL (VHSIC Hardware Description Language) is a hard
used in electronic design automation to describe digita
such as field-programmable gate arrays and integrated
used as a general purpose parallel programming langua

https://en.wikipedia.org/wiki/VHDL

Summary: History of VHDL

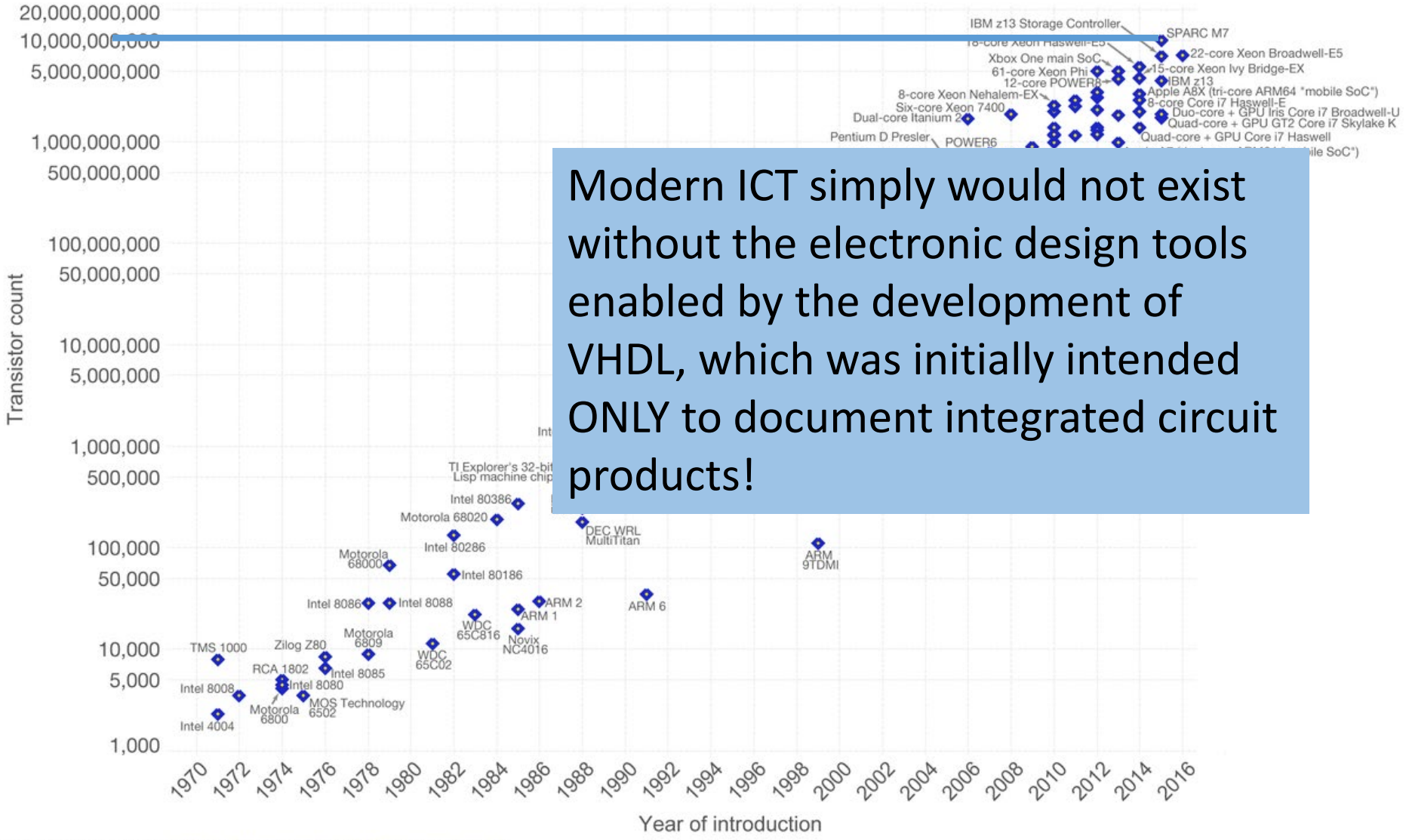
1981 Initiated by US DoD to address hardware life-cycle crisi
1983-85 Development of baseline language by Intermetri
1986 All rights transferred to IEEE
1987 Publication of IEEE Standard
1987 Mil Std 454 requires comprehensive VHDL description
1994 Revised standard (named VHDL 1076-1993)
2000 Revised standard (named VHDL 1076 2000, Edition)
2002 Revised standard (named VHDL 1076-2002)
2007 VHDL Procedural Language Application Interface stand
2009 Revised Standard (named VHDL 1076-2008)

https://www.doulos.com/knowhow/vhdl_designers_guide/a_brief_history_of_vhdl/



Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Modern ICT simply would not exist without the electronic design tools enabled by the development of VHDL, which was initially intended ONLY to document integrated circuit products!

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Logistics centers designed for the Physical Internet

Transit Center facilitating the truck-to-truck transshipment of trailers along relays networks through the Physical Internet

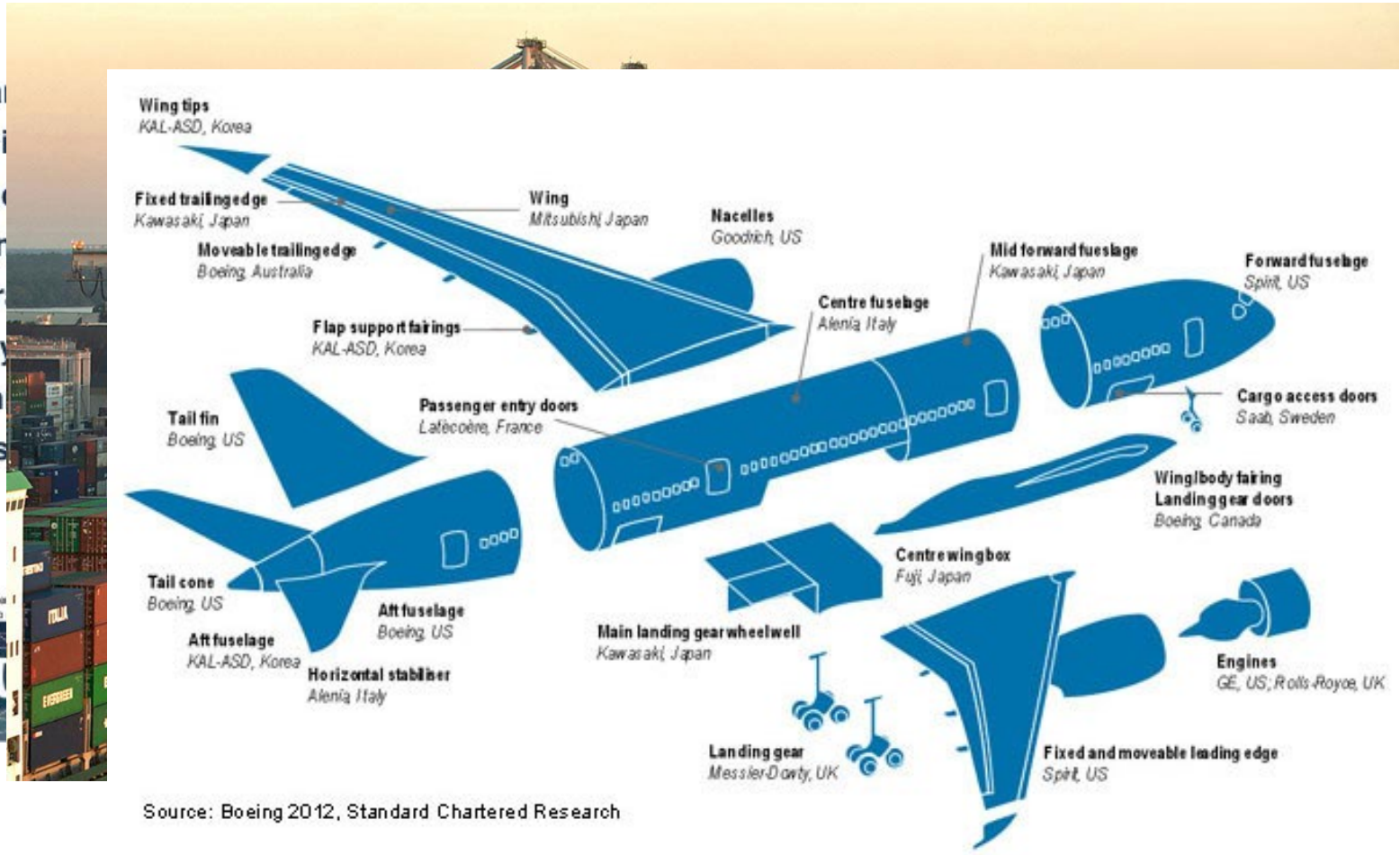
Reference: Meller, R.D., B. Montreuil, C. Thivierge &Z. Montreuil (2012), *Functional Design of Physical Internet Facilities: A Road-Based Transit Center*, in Progress in Material Handling Research: 2012, MHIA, Charlotte, NC, to appear (2012).

Logistics centers designed for the Physical Internet

Tra
faci
truc
tran
of tr
relay
th
Phys

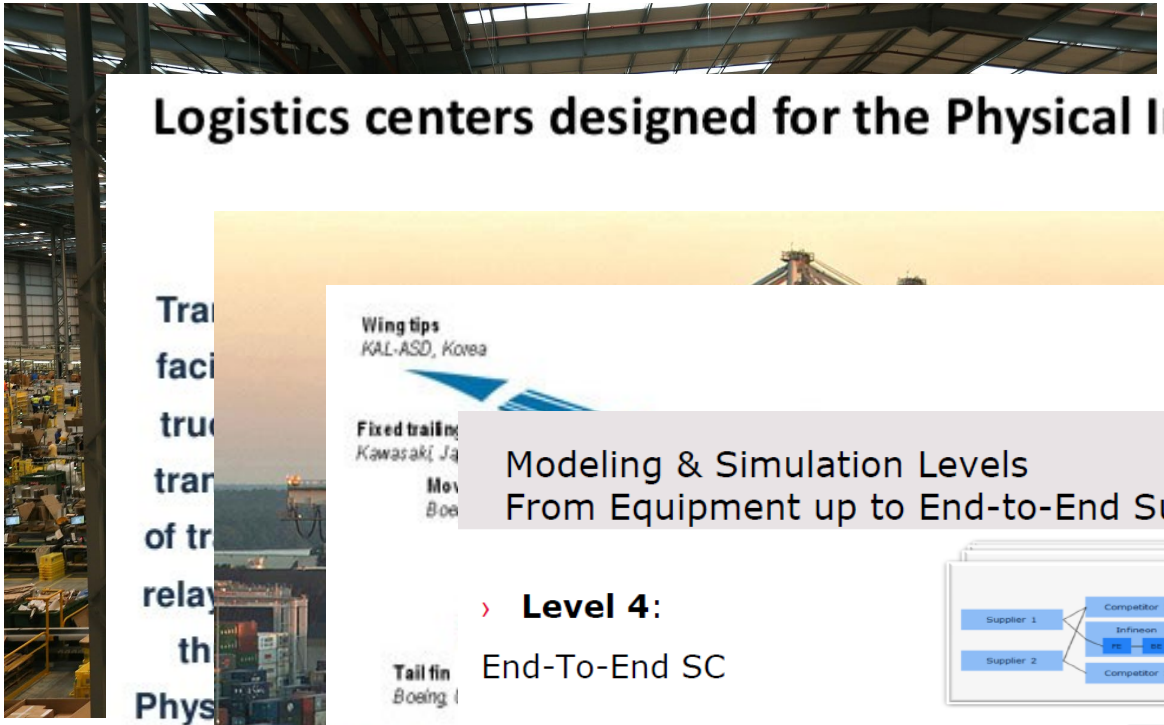
Reference : Mollet
Charlotte, NC, to

Logistics centers designed for the Physical Internet



Source: Boeing 2012, Standard Chartered Research

Logistics centers designed for the Physical Internet



Wing tips
KAL-ASD, Korea

Fixed trailing
Kawasaki, Ja

Mov
Boe

Tail fin
Boeing

Tail cone
Boeing, US

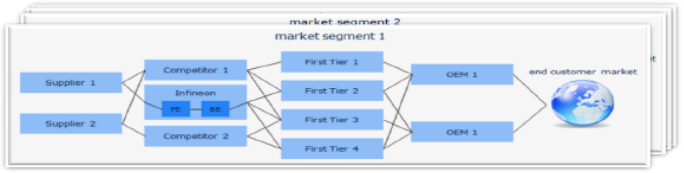
Aft fu
KAL-

Source: E

Modeling & Simulation Levels
From Equipment up to End-to-End Supply Chain

› **Level 4**:

End-To-End SC



› **Level 3**:

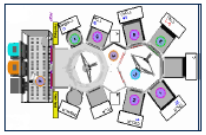
Semiconductor company SC

› **Level 2**:

Factory Supply Chain SC

› **Level 1**:

Cluster/Equipment SC



What the world needs

- Ability to design (control, predict, specify) the large scale socio-technical, cyber-physical systems that sustain modern life, because they enable the ***flow of material through networks of resources that transform that material to higher value***

- This is our domain!

Fundamental Truths

- Operational control of these systems does not exist without material handling
- The physical processes of material handling, *vis-à-vis control*, often are not considered explicitly in systems design—they are effectively an afterthought

OPPORTUNITY!

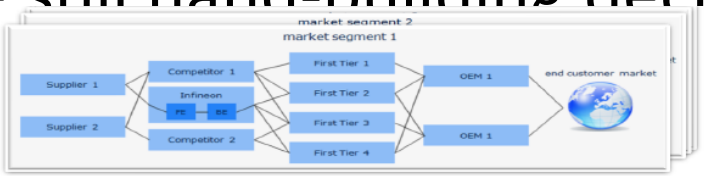
Fundamental Truths

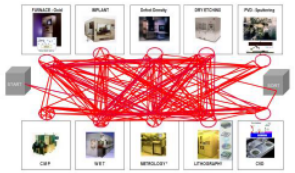
- By and large, we are still hand-building decision support analysis models *that we already know how to build!*
- As a result, we are too slow, too expensive (and not well integrated with other decisions) so we are not used well
- And our analysis methods are becoming commodities

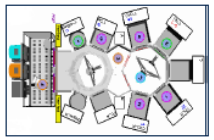
Even if we are happy to work at the individual process level, we will have a lot of competition

Fundamental Truths

- By and large, we are still hand-building decision
  - **Level 4**: End-To-End SC
  - **Level 3**: Semiconductor company SC
  - **Level 2**: Factory Supply Chain SC
  - **Level 1**: Cluster/Equipment SC







how

d not

not

Even if we are happy to work at the individual process level, we will have a lot of competition

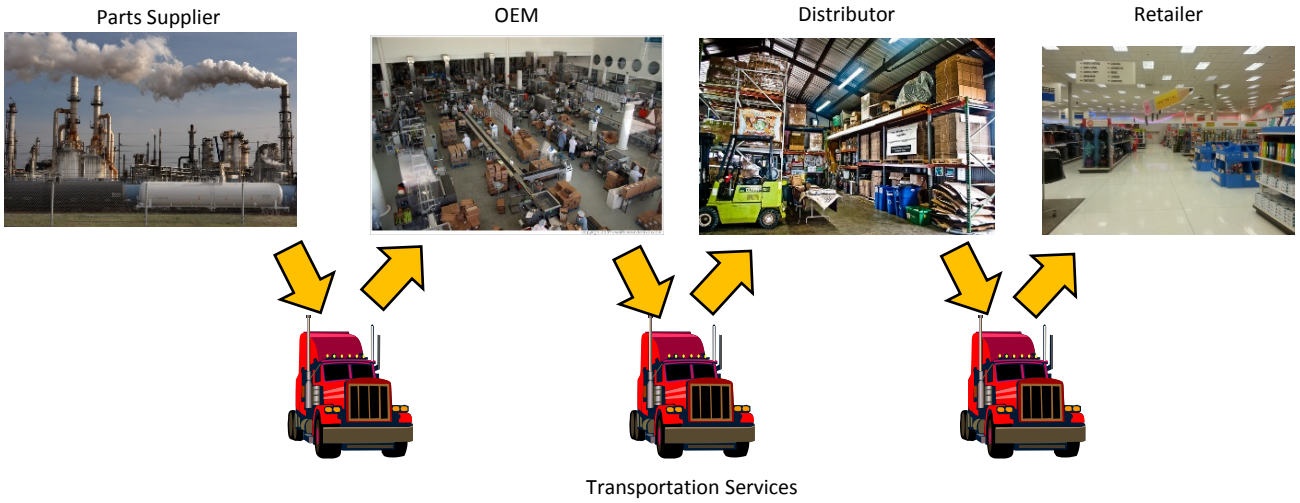
Conclusion

- Future material handling research, as we know it, must have a systems perspective, must integrate decisions about MH system design and operational control with other systems-level design and operations decisions
  - Domain experts will always know more than we do about the system of interest
  - We *must* know something that is not a commodity, that contributes effectively to systems design and operational control decision making
  - We *must* translate our results into implementable recommendations

How might we reach that future?

- Shared, unifying framework for system specification
- *Lingua franca* for system specification (a la VHDL)
- Specification authoring and editing tools
- Framework for decision-making, both for system design and system control
- Decision support analysis automation
- Decision support analysis integration with system specification

Discrete Event Logistics Systems

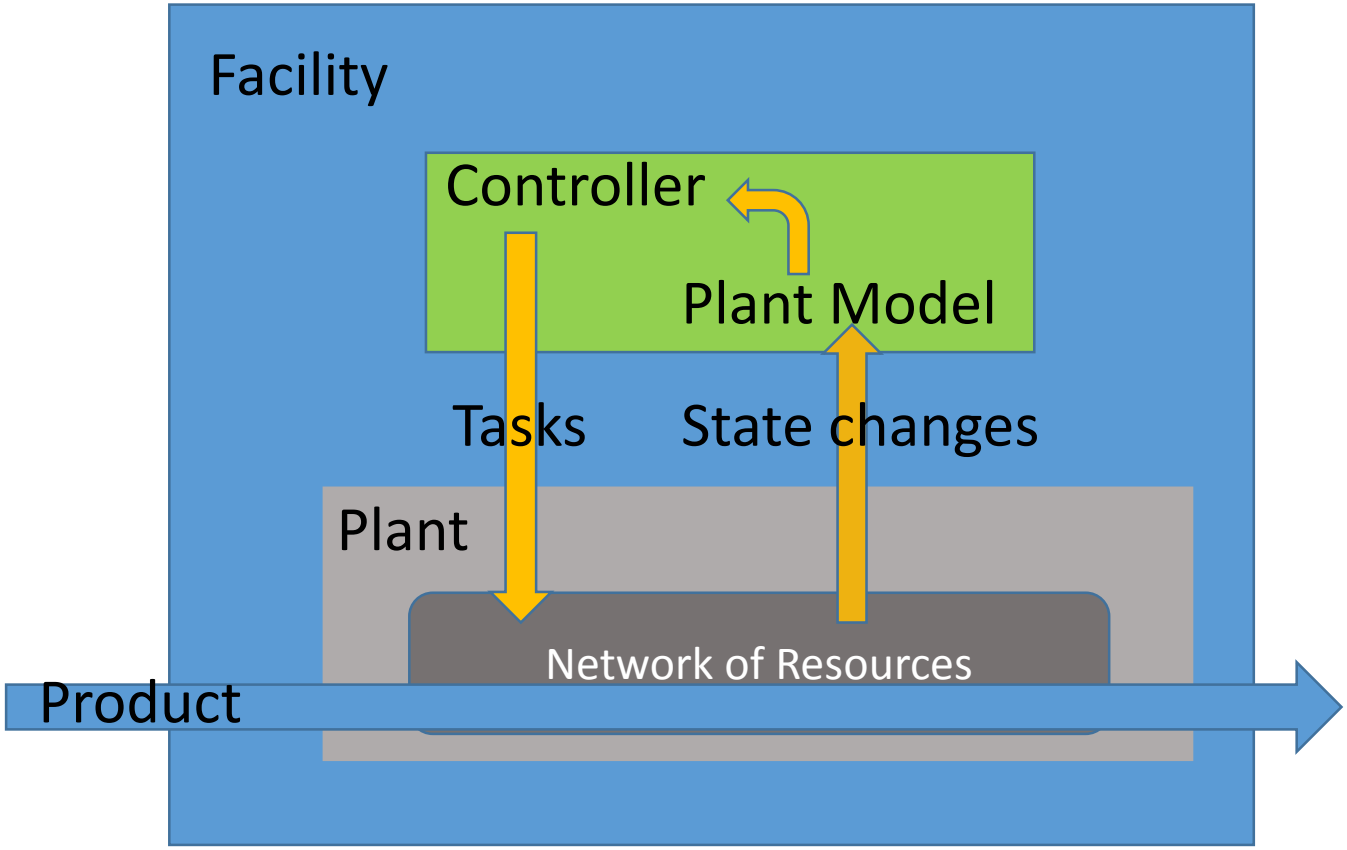


Units of flow move through a network of resources, which execute processes that transform the units of flow in some way—location, age, configuration, information, etc. These are **"discrete event logistics systems" or DELS**.

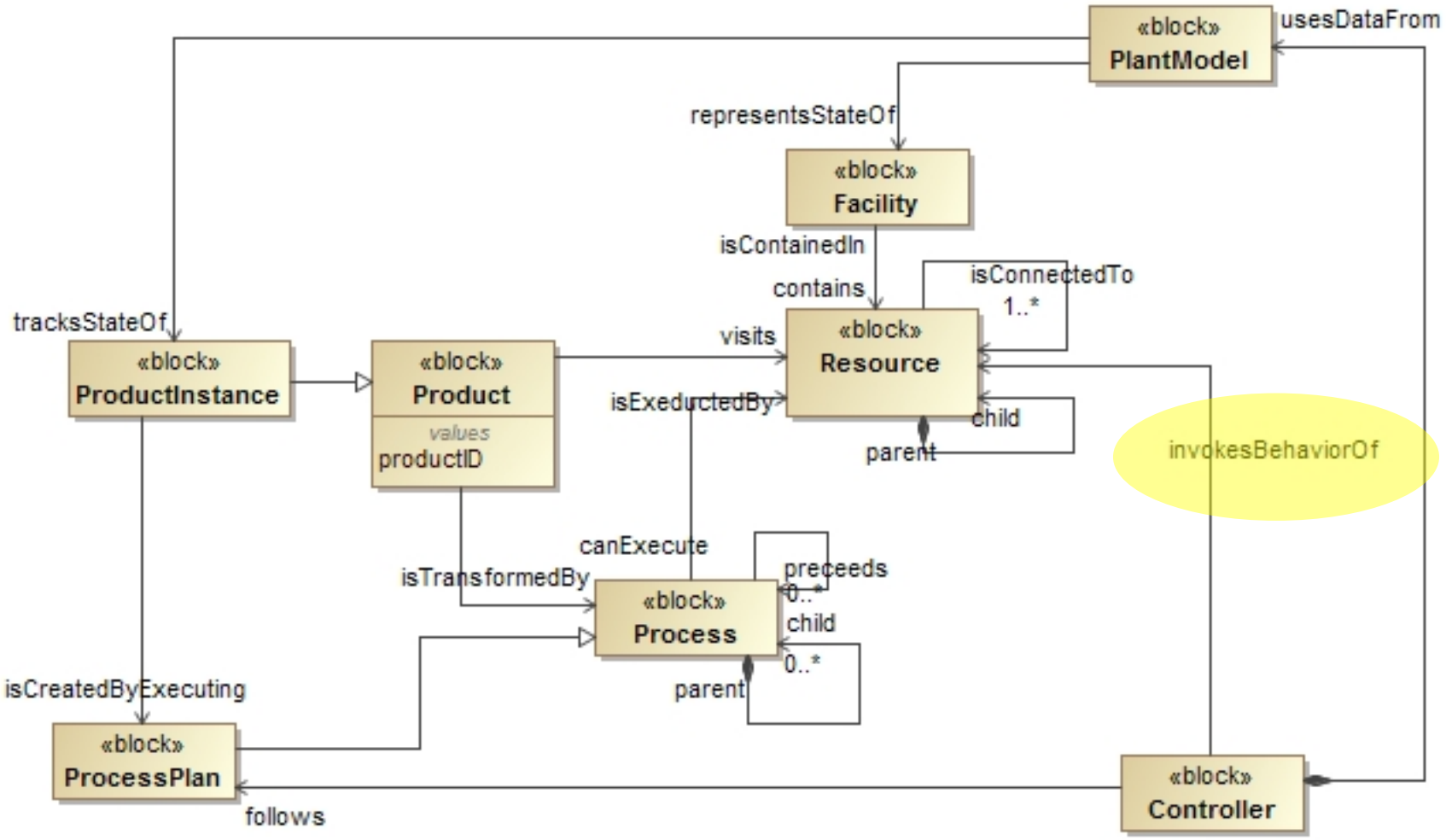
Transformations can be adequately described by their start and end events, and by the summary description of the state change accomplished.

Toward a language for DELS

- Product
- Process
- Resource
- Facility
- Control
- Plant Model

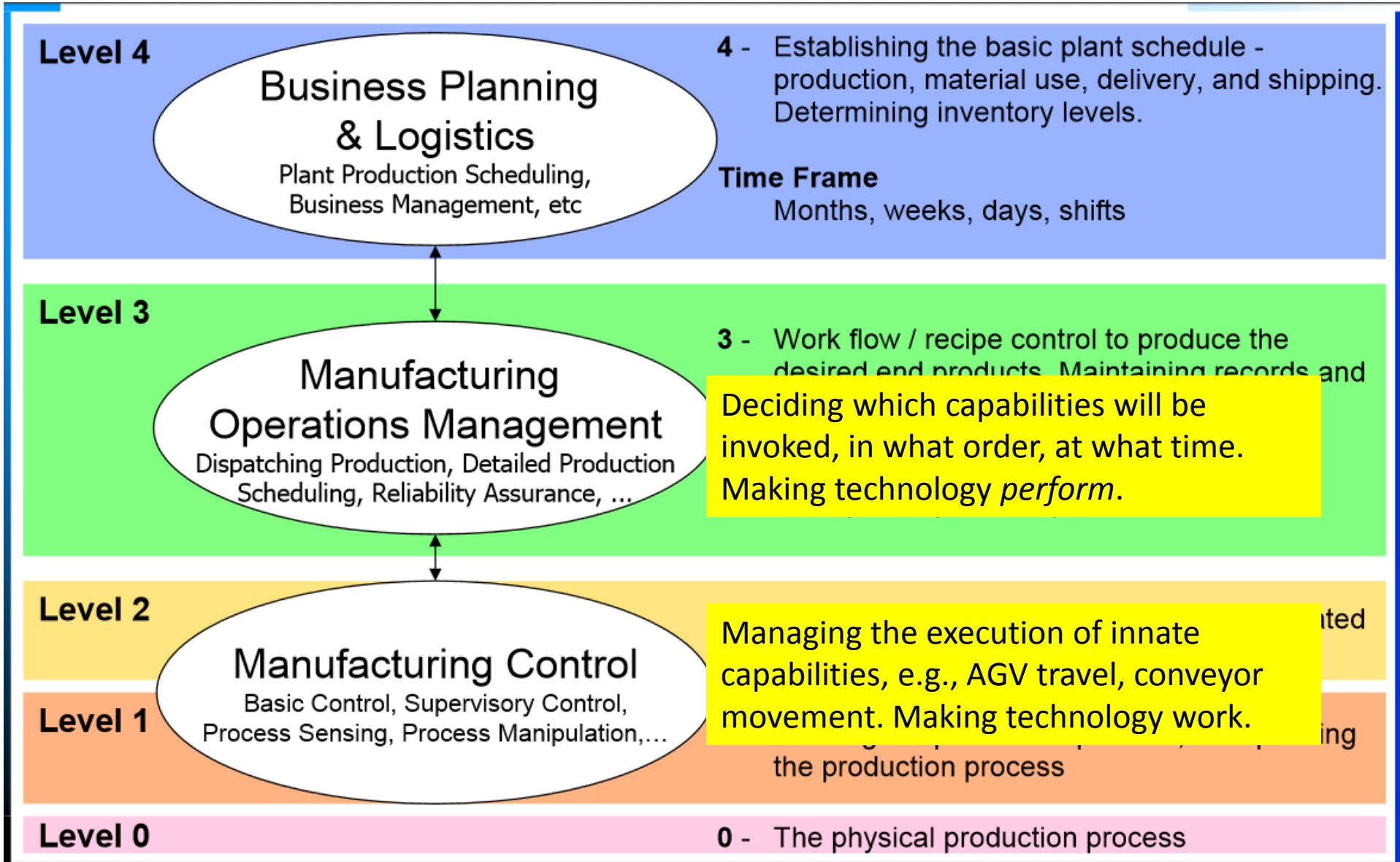


Resource capabilities: make, move, store (get, put)


«block»
PlantModel
usesDataFrom
representsStateOf
«block»
Facility
isContainedIn
contains
isConnectedTo
1..*
tracksStateOf
visits
«block»
Resource
«block»
ProductInstance
«block»
Product
values
productID
isExeductedBy
child
parent
invokesBehaviorOf
canExecute
isTransformedBy
preceeds
0..*
«block»
Process
child
0..*
parent
isCreatedByExecuting
«block»
ProcessPlan
«block»
Controller
follows

ANSI/ISA-95

Purdue Enterprise Reference Architecture (PERA)

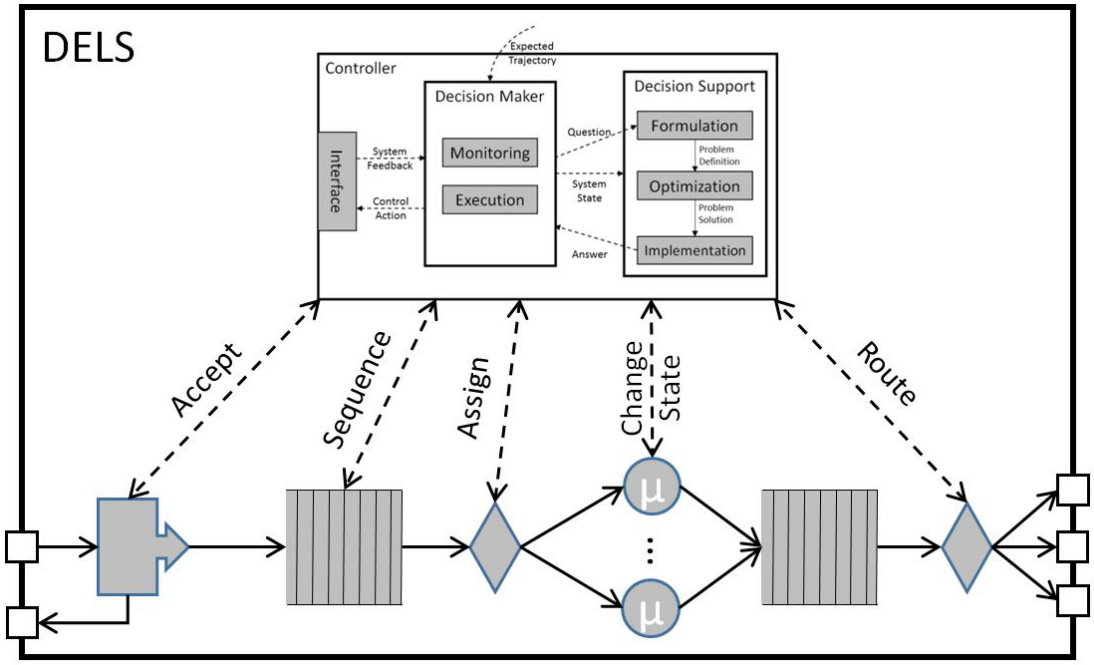


Operational control decisions are implemented by moving material!

"Make" is not an operational control decision!

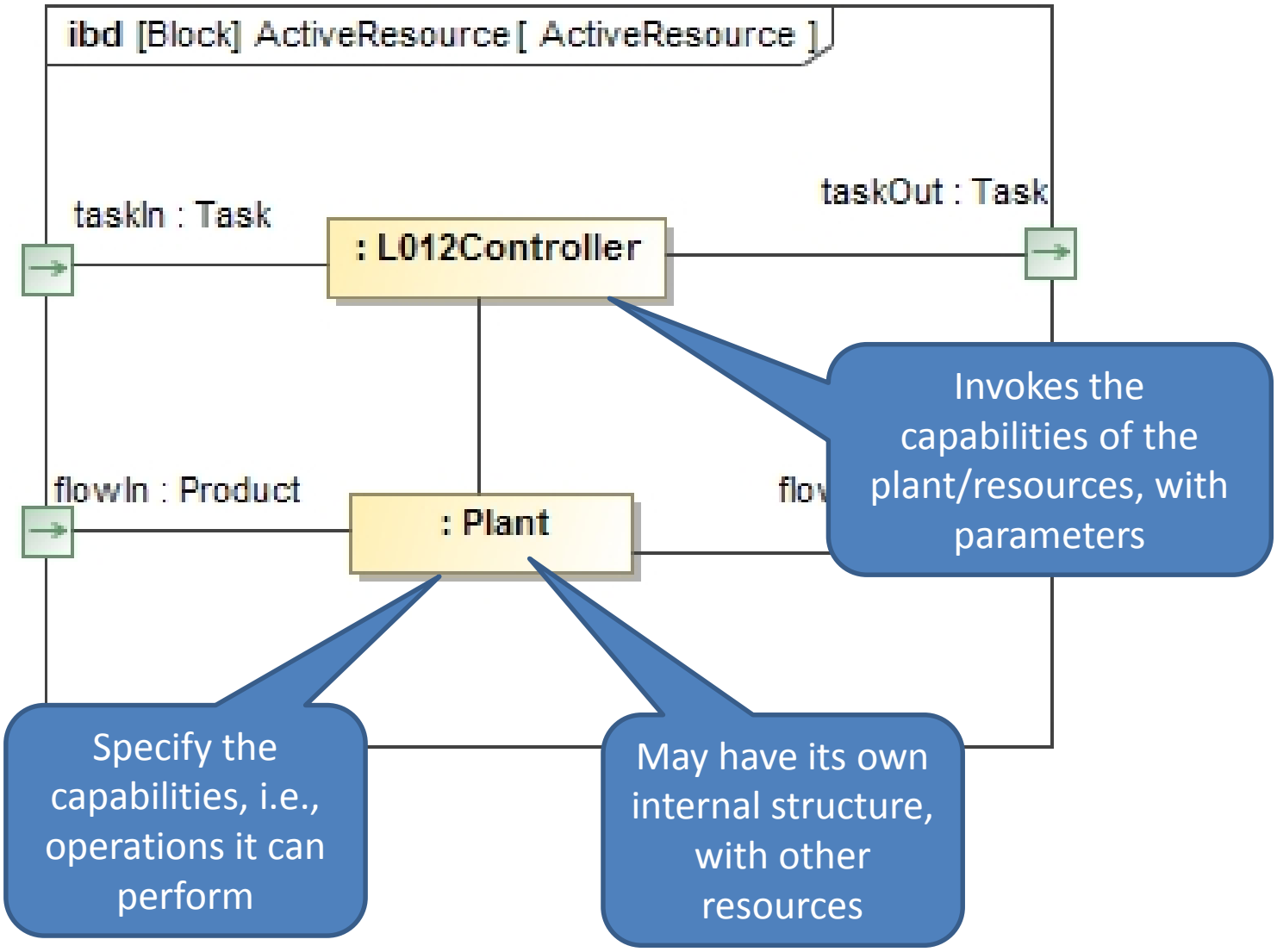
OPERATIONAL CONTROL QUESTIONS

Control questions provide a mapping from a formal functional definition of control activities for DELS to formal (math programming) analysis models.



- 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)

PLANT/CONTROL SEPARATION



A motivating example

A central fill pharmacy (CFP) is “a pharmacy which is permitted by the state in which it is located to prepare controlled substances orders for dispensing pursuant to a valid prescription transmitted to it by a registered retail pharmacy and to return the labeled and filled prescriptions to the retail pharmacy for delivery to the ultimate user” (21 CFR 1300.01 (44) [Title 21 Food and Drugs; Chapter II Drug Enforcement Administration, Department of Justice; Part 1300 Definitions]).

Scale and Scope

- Thousands of drugs (associated Pareto for orders)
- 30,000 scripts per day
- Several hundred to a thousand local pharmacies (with associated Pareto for orders)

High speed filling into vials in pucks

High flexibility robotic filling into vials deposited in totes

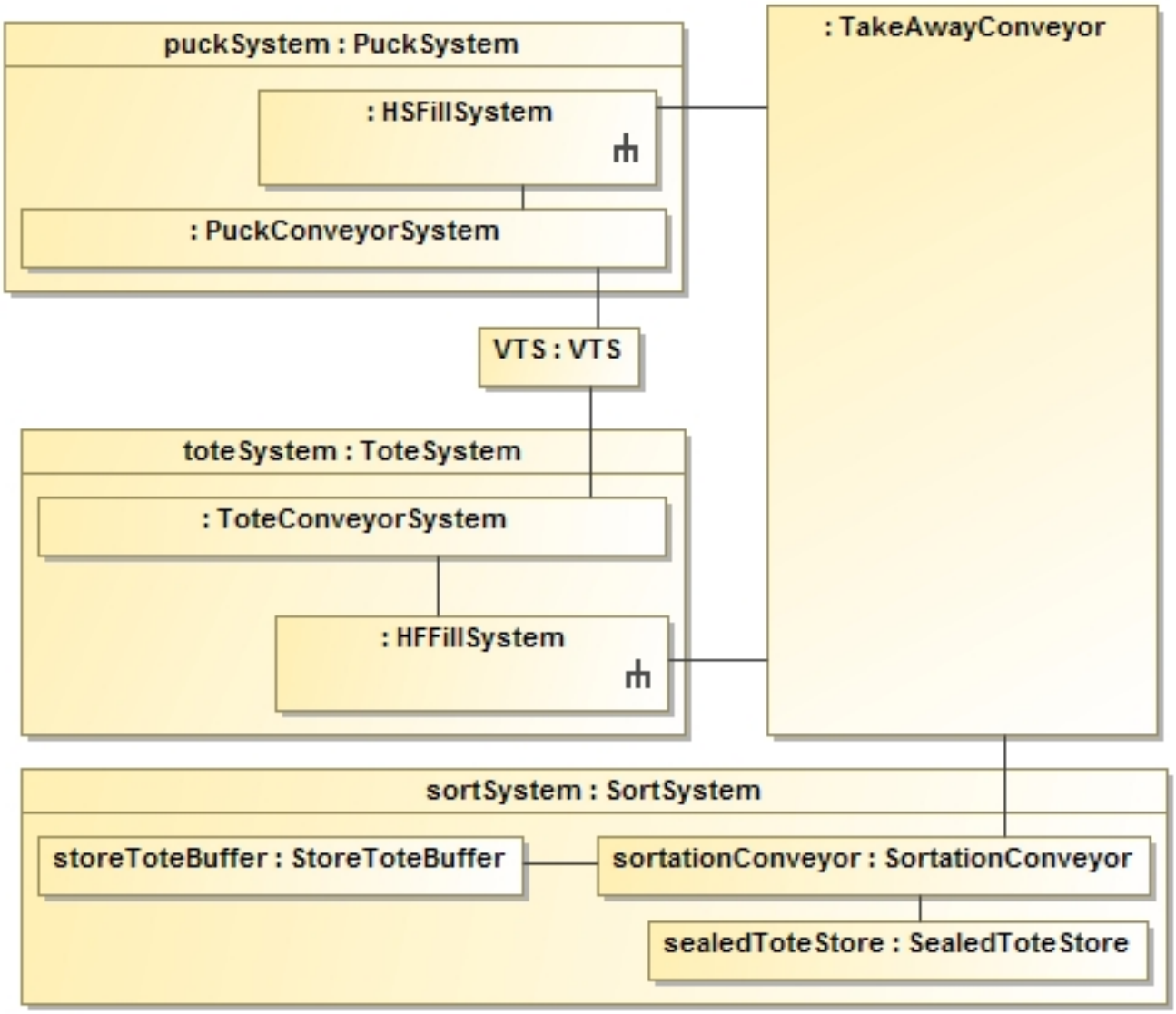
Unit of use

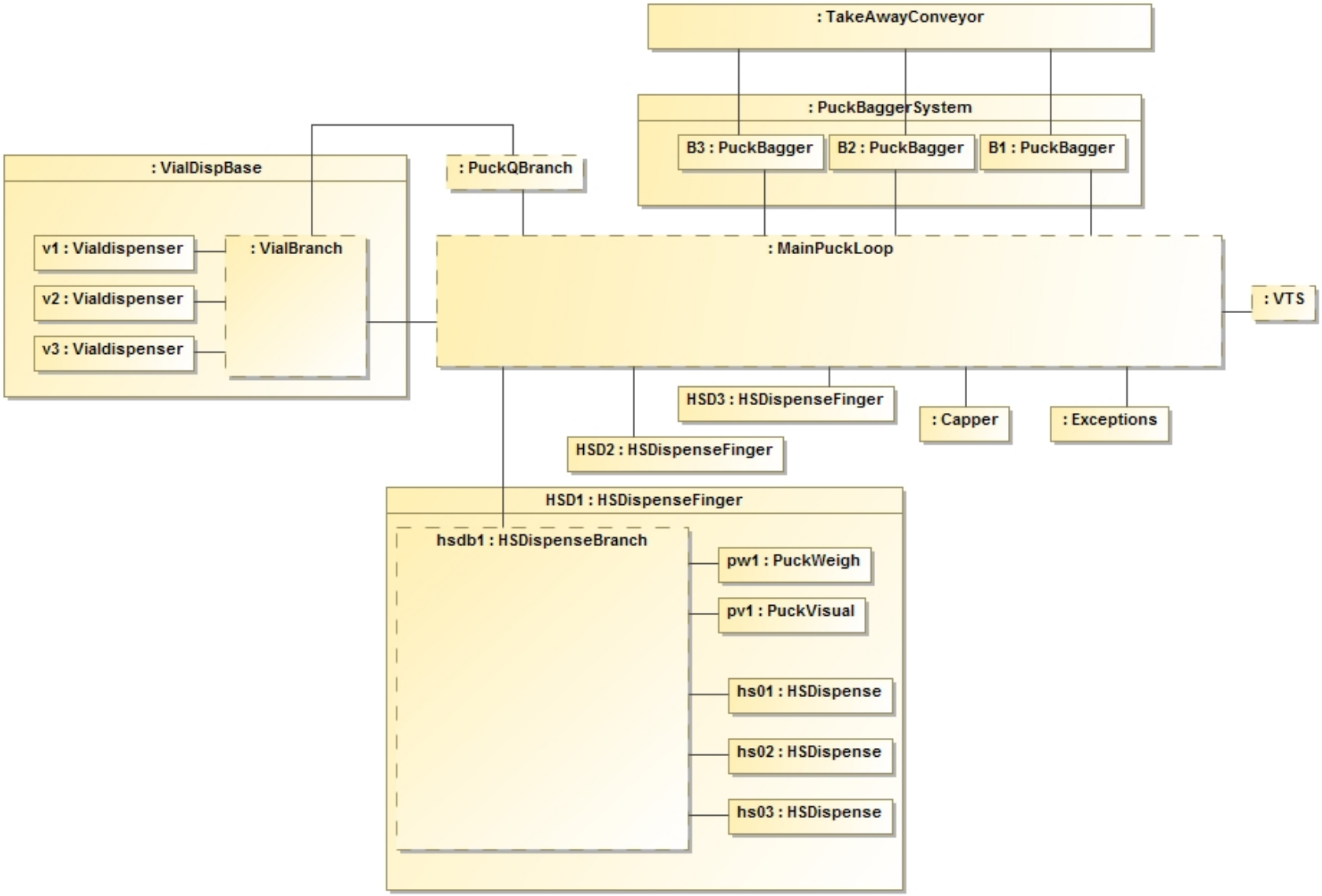
Bagging

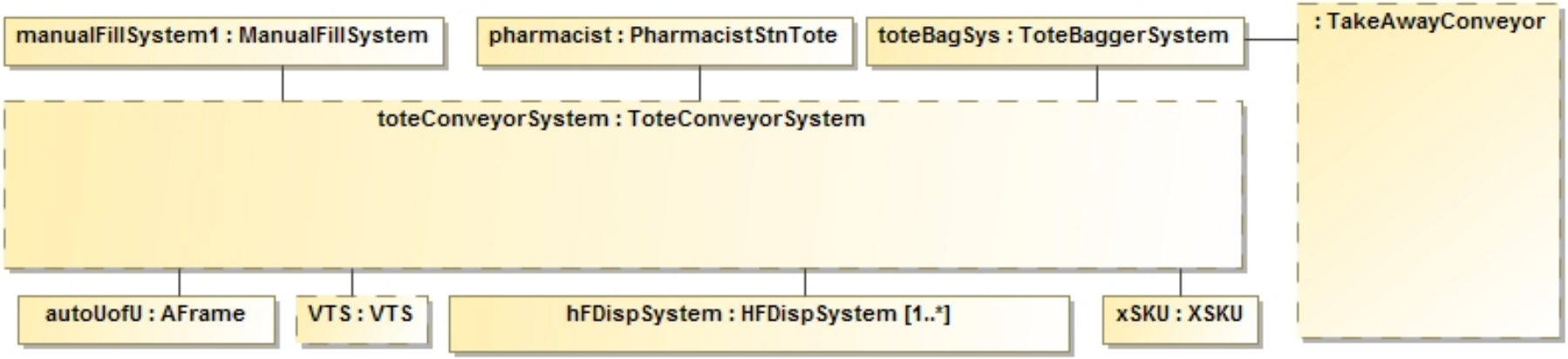
Sorting

Product

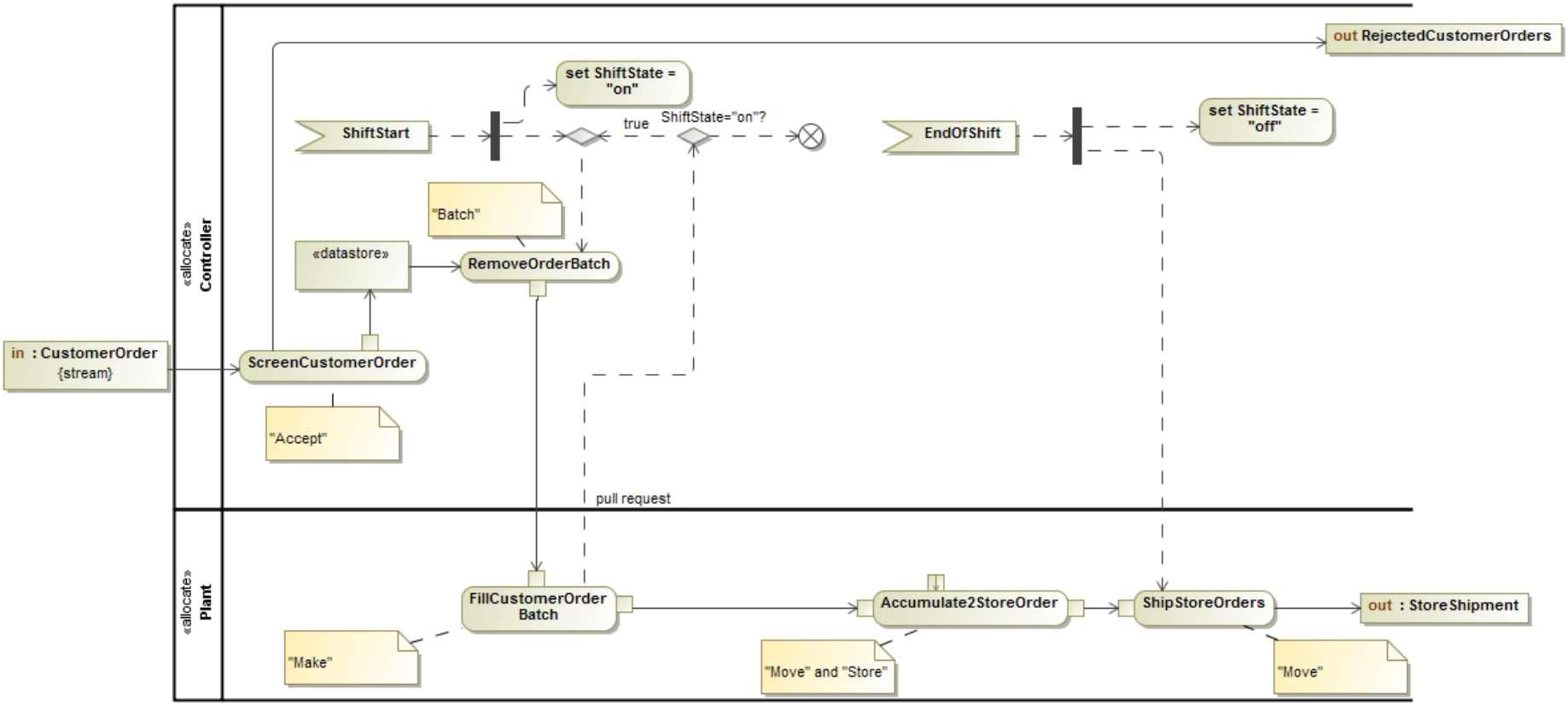
- The product of a CFP is a *set* of assembled orders ready to ship to the originating local pharmacy
  - Customer order may have one or more scripts
  - Scripts may be automatically dispensed from either the puck line or the tote line
  - Scripts may be manually dispensed from the tote line
  - An order must contain only scripts for drugs that are available in the CFP
  - All the orders for a local pharmacy must be collected for shipment overnight

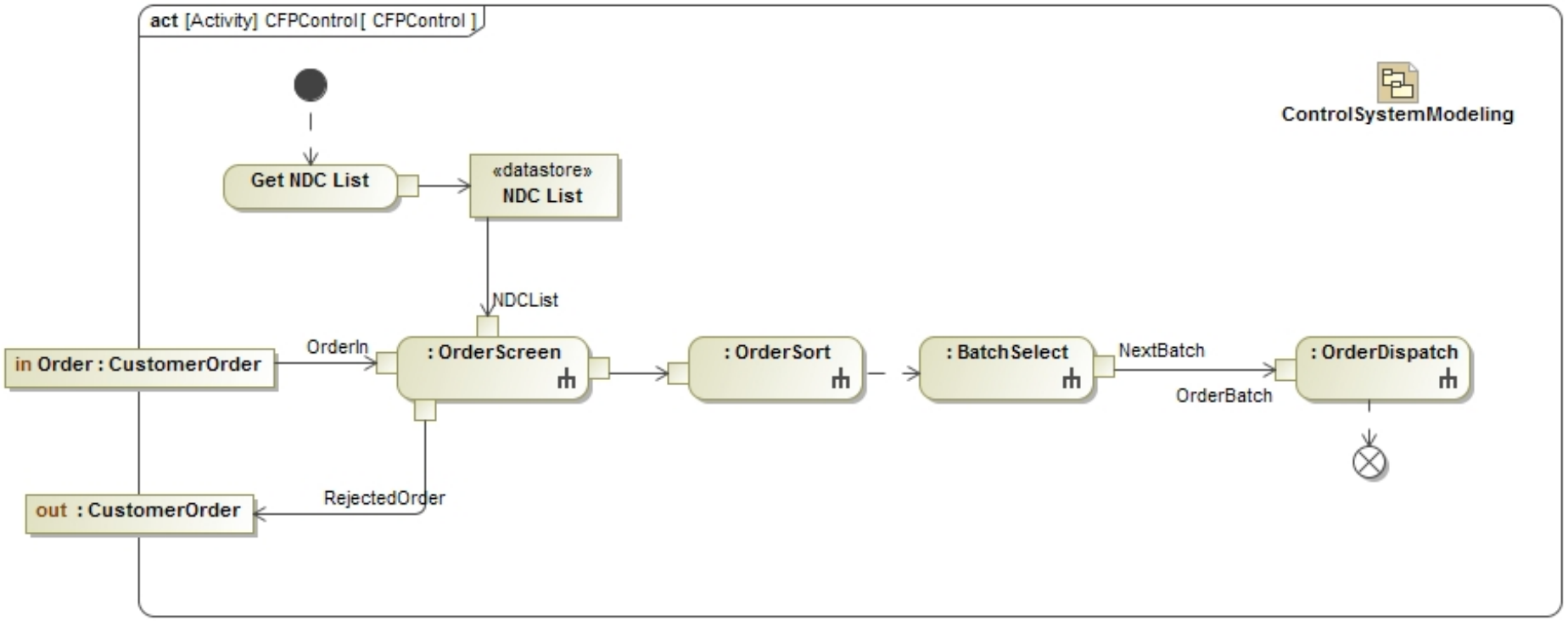

puckSystem : PuckSystem
: HSFillSystem
: PuckConveyorSystem
: TakeAwayConveyor
VTS : VTS
toteSystem : ToteSystem
: ToteConveyorSystem
: HFFillSystem
sortSystem : SortSystem
storeToteBuffer : StoreToteBuffer
sortationConveyor : SortationConveyor
sealedToteStore : SealedToteStore

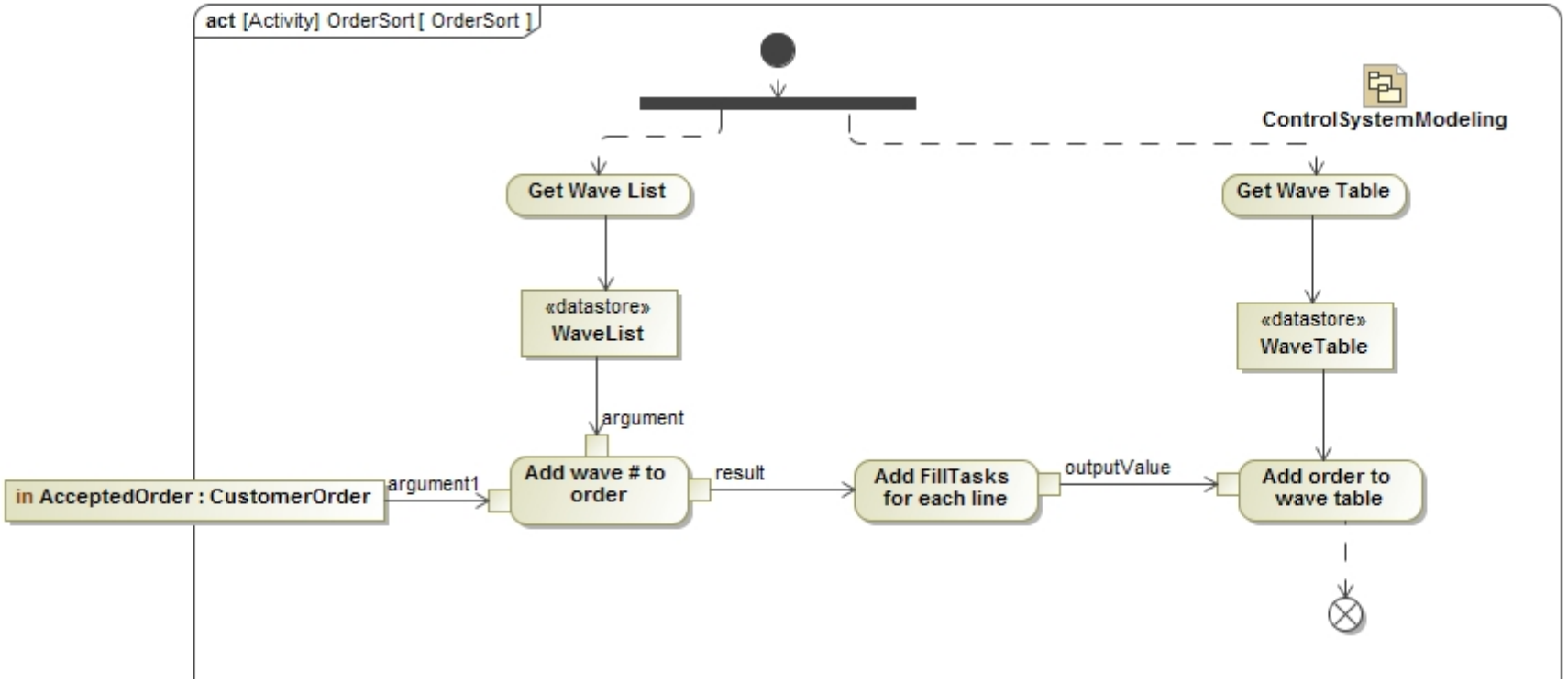

: TakeAwayConveyor
: PuckBaggerSystem
B3 : PuckBagger
B2 : PuckBagger
B1 : PuckBagger
: VialDispBase
: PuckQBranch
v1 : Vialdispenser
v2 : Vialdispenser
v3 : Vialdispenser
: VialBranch
: MainPuckLoop
: VTS
HSD3 : HSDispenseFinger
: Capper
: Exceptions
HSD2 : HSDispenseFinger
HSD1 : HSDispenseFinger
hsdb1 : HSDispenseBranch
pw1 : PuckWeigh
pv1 : PuckVisual
hs01 : HSDispense
hs02 : HSDispense
hs03 : HSDispense

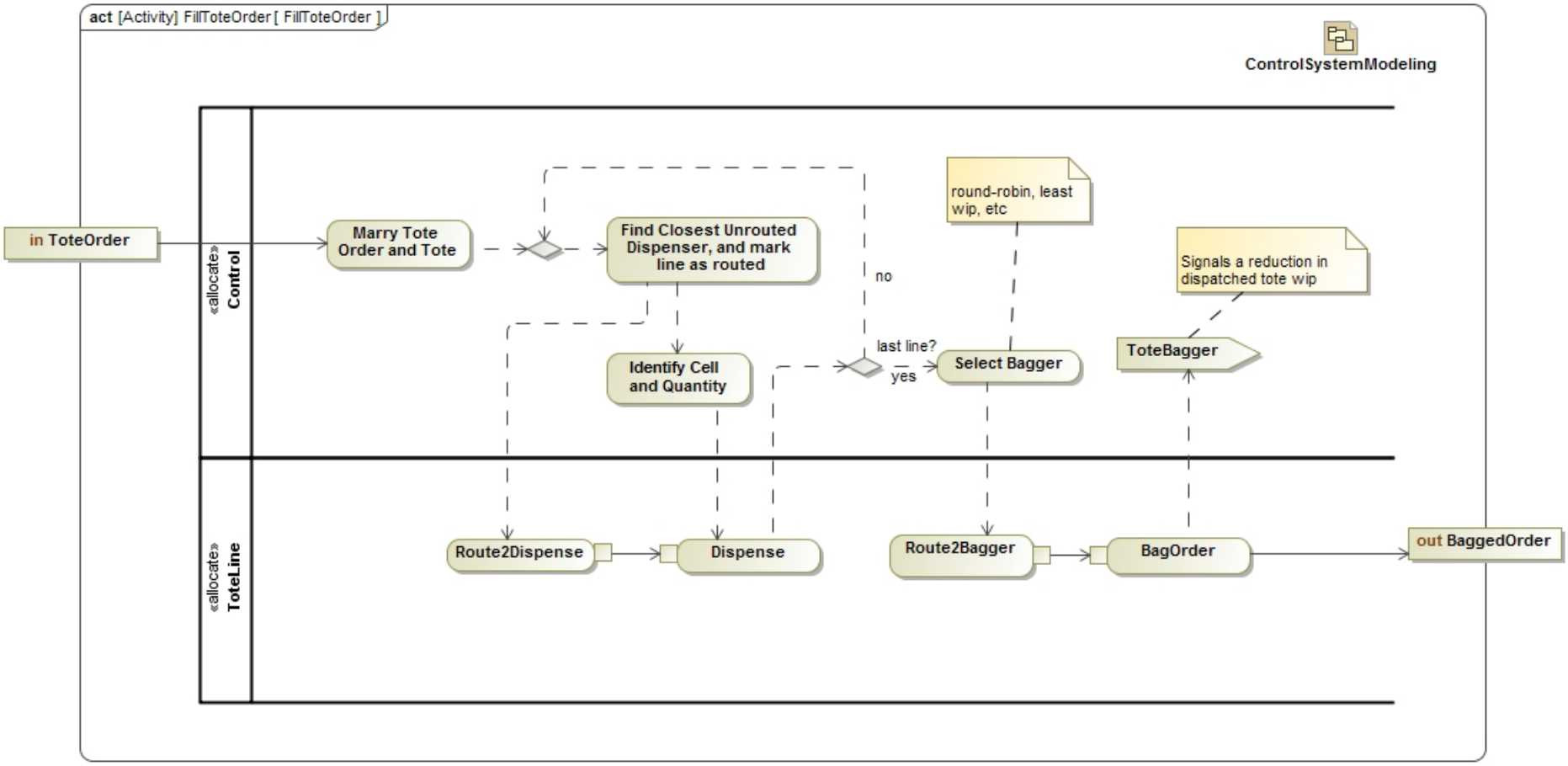

manualFillSystem1 : ManualFillSystem
pharmacist : PharmacistStnTote
toteBagSys : ToteBaggerSystem
: TakeAwayConveyor
toteConveyorSystem : ToteConveyorSystem
autoUofU : AFrame
VTS : VTS
hFDispSystem : HFDispSystem [1..*]
xSKU : XSKU

Basic processes




act [Activity] CFPControl[CFPControl]
ControlSystemModeling
Get NDC List
«datastore»
NDC List
NDCList
in Order : CustomerOrder
OrderIn
: OrderScreen
: OrderSort
: BatchSelect
NextBatch
: OrderDispatch
OrderBatch
out : CustomerOrder
RejectedOrder


act [Activity] OrderSort [OrderSort]
ControlSystemModeling
Get Wave List
Get Wave Table
«datastore»
WaveList
«datastore»
WaveTable
argument
in AcceptedOrder : CustomerOrder
argument1
Add wave # to order
result
Add FillTasks for each line
outputValue
Add order to wave table


act [Activity] FillToteOrder [FillToteOrder]
ControlSystemModeling
in ToteOrder
«allocate» Control
Marry Tote Order and Tote
Find Closest Unrouted Dispenser, and mark line as routed
round-robin, least wip, etc
Signals a reduction in dispatched tote wip
no
last line?
yes
Identify Cell and Quantity
Select Bagger
ToteBagger
«allocate» ToteLine
Route2Dispense
Dispense
Route2Bagger
BagOrder
out BaggedOrder

Why is this worth doing?

- No ambiguity about the system concept or architecture
- Ability to specify any variation of this basic “style” of CFP
- Single source of truth for analyses to support all planning and operational control
- Specification for simulation model that enables experimentation with control rules and parameter settings

Conclusions

- It’s worth doing
- It’s really hard (at first), but so was DES and optimization
- We are gaining experience and fleshing out a hierarchical abstraction

- I don’t see another way for us to get where we need to be—remember VHDL!

Thank you
and
Safe Travels Home!