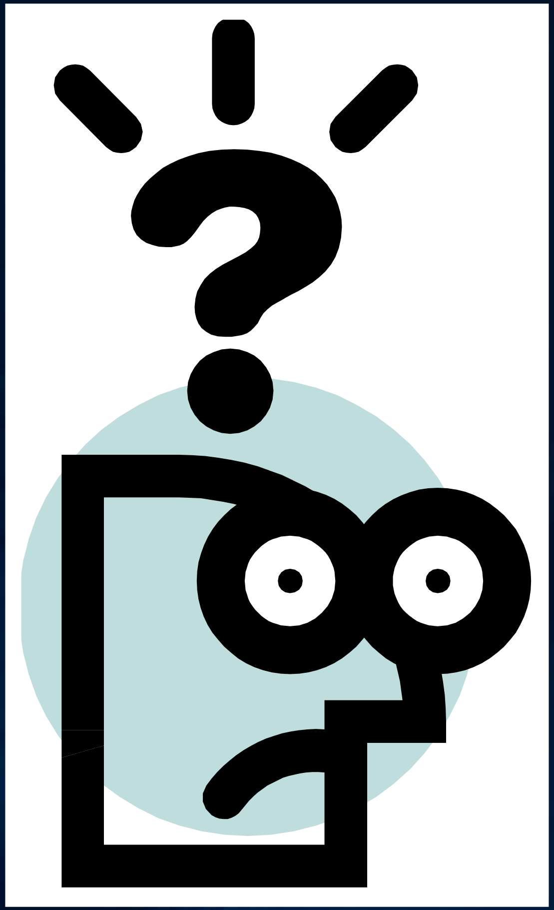
Teaching System Design to Undergrads

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

Your turn



How can we teach “applied” courses if we’ve never been in practice?

- Inventory, production planning, logistics, manufacturing systems, …

- We teach principles, methods and tools
- We can do the same thing for system design
- So what are those principles, methods, and tools?

System Design Problem Example

- Pick any CICMHE design competition case!
- I'll use the most recent one in this presentation—Fast Foods Now—an abridged version of which was emailed to you on Sunday

Fast Foods Now

- Warehousing Scenario
  - SKU master
  - Order data
  - Description of goods flow, storage constraints, growth projections, etc
- Requirement: design a new warehouse
  - Channel
  - Storage systems
  - Picking technology

This is a very typical "IE" systems design problem.

Engineering design of systems requires answering the following questions (not in a single sequence!):

- What is the *context* for this system?
  - How does this system create *value*?
- What *functions* must this system provide/perform?
- How can we best organize these functions into an *architecture*?
- What *technologies* should we choose to enable these functions?
- How shall we *size and configure* the technologies?
- What shall be the final *embodiment* of the system?
- ***How will we judge whether or not this system is a good system?***

As we will see, there usually is some iteration involved!

Theme for today: How can we help students learn to answer these questions?

Some “axioms”:

- What you can learn depends a lot on what you already know
- For these questions, there may be lots of answers, both good and bad
- Answering these questions well and quickly requires knowledge, experience, methods and tools
- Discernment is not learned passively
- The novice system designer needs a process to follow
- The expert system designer may use a different process
- This is an “interactive” session!

Decision Process (a “methodology”)

- Context identified
- Flows identified
- Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated
- Iteration

System Design Process

- Starts by identifying *functions*
- Ends by specifying:
  - Processes performed to transform input products to output products
    - Production processes are "make" processes identified in the design of the product
    - Logistics processes are the move and store/retrieve processes necessary to enable the production processes
  - Resources that will execute processes
    - The behaviors these resources must be capable of exhibiting
  - Facility
    - How the resources will be organized

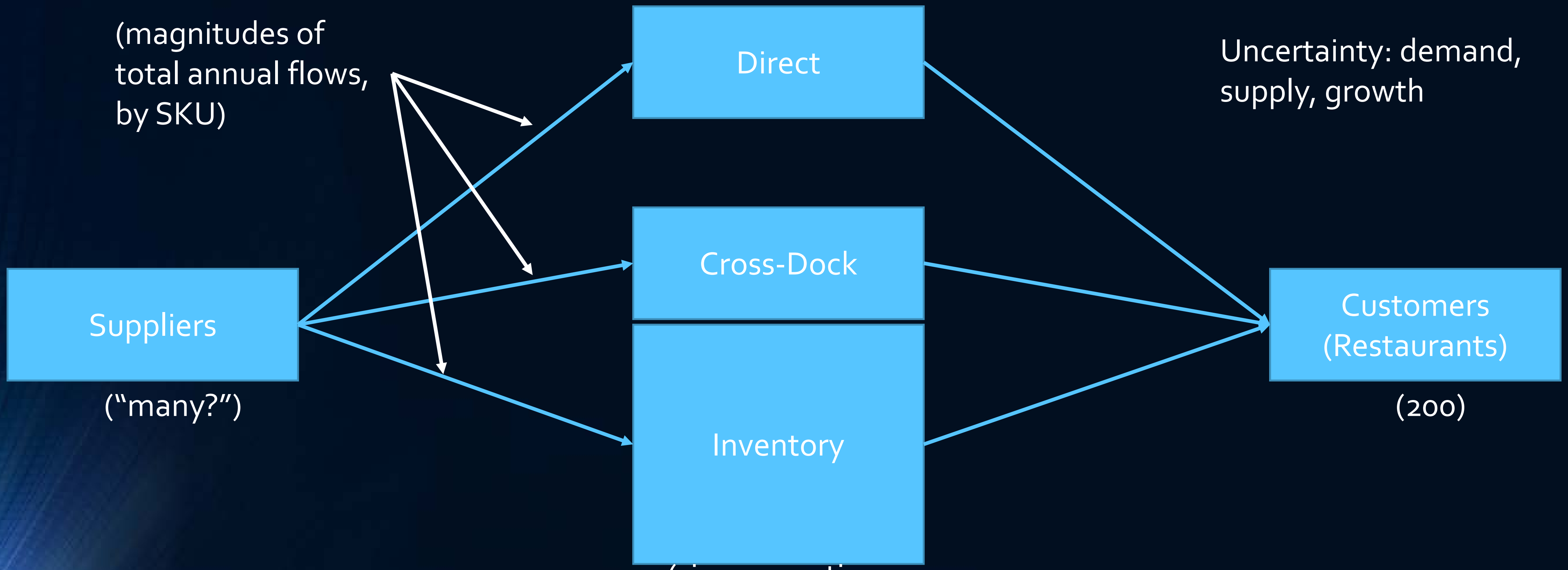
Context

- Take 5 minutes to answer these two questions.
  - What is in the context for this design problem?
  - What is out of scope?
- Focus on:
  - What is “flowing”?
  - Who/what are the “actors”?
  - What do the actors “do”?

As-Is Scenario

(magnitudes of total annual flows, by SKU)

Uncertainty: demand, supply, growth



("many?")

(200)

(size, operating characteristics)

Within limits, FFN can choose which channels are appropriate for each SKU

In Scope

- How to select channel for each SKU
- How to set inventory policy for SKUs in inventory
- Seasonality
- How to integrate operations of cross-dock and order-picking operations spread across two distinct warehouse facilities
- Selection, sizing and configuration of storage technologies
- Selection of order-picking method(s)

Out of Scope

- WMS
- Yard management
- Outbound truck loading
- Routing deliveries to restaurants

As seen by students with no experience, thus no starting point for understanding

How Can We Help Them?

- System semantics:
  - P/P/R/F and structure/behavior/control and function
- Domain semantics
  - Supplier, Customer, Flow, Unit of Handling, Channel
- Domain Processes
  - Transportation, Cross-Docking, Receiving, Inventory, Shipping
- Principles
  - Conservation of flow—over some time interval!
  - Uncertainty
- Use the as-is scenario as the platform for introducing the domain knowledge

System Semantics (1)

- Product: what is flowing
- Process: how the flow is transformed
- Resource: that which executes the transformation
- Facility: the organization of resources through which the product flows

These can be directly observed in any particular system domain, any application domain.

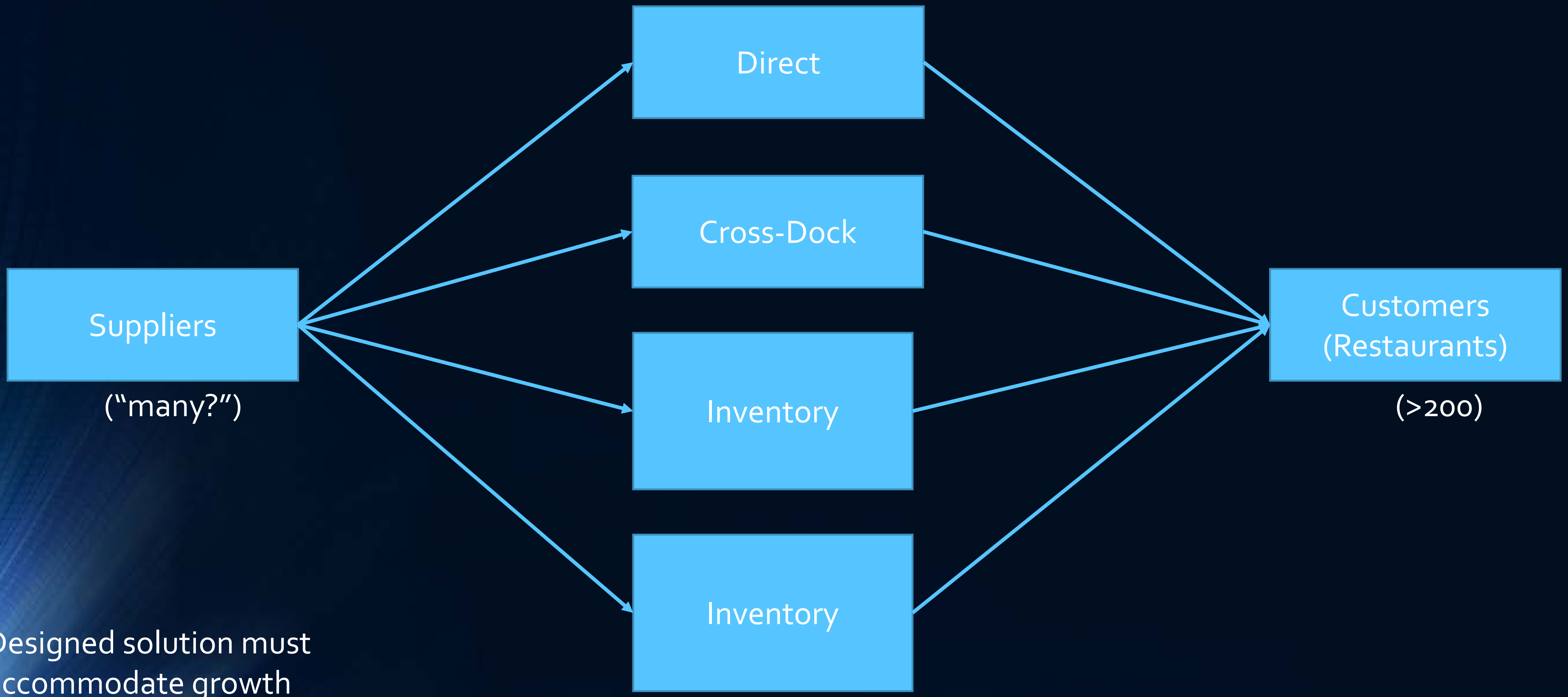
System Semantics (2)

- Structure: the set of resources and how they are connected (this determines what flows are possible)
- Behavior: a resource's possible behaviors are described by the kinds of processes it is capable of performing
- Control: determines which resources perform which processes on a unit of flow and the sequence of processes executed by a resource

System Semantics (3)

- Function
  - An abstraction of process, with no specific associated product or resource
  - Describes a *type* of process that might be applied to a *type* of product using some *type* of resource
- Examples:
  - In warehousing, we can speak of an *inbound function* without identifying a specific warehouse, specific products, specific processes or specific resources
  - In healthcare delivery, we can speak of *triage* without identifying a specific scenario
  - In manufacturing, we can speak of *work in process buffer* without identifying a specific product

To-Be Scenario



Designed solution must accommodate growth

How might we identify the "best" warehouse design for FFN?

Minimize the cost to meet a service level of $(1-\alpha)100\%$.

What System *Functions* Are Essential for Creating Value?

- Context identified
- Flows identified
- Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated
- Iteration

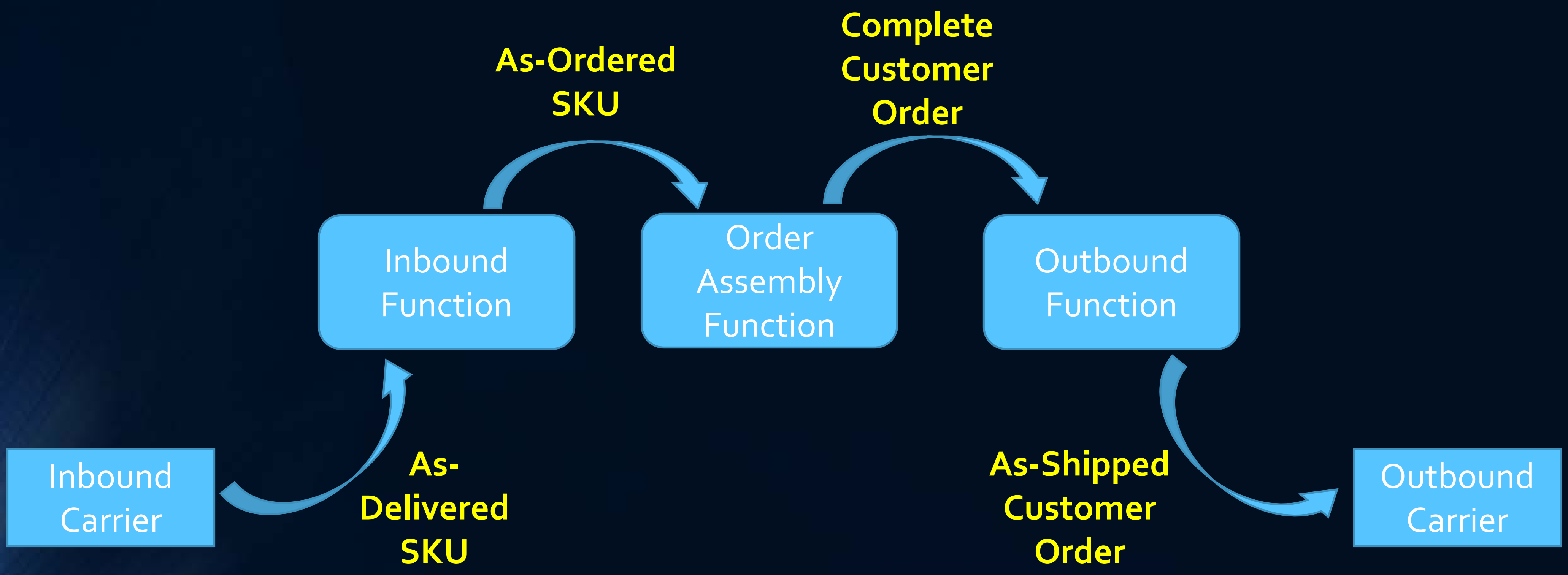
- Follow one particular SKU through the warehouse channel and record every *process* executed by a *resource*
- Imagine you have a GoPro camera on a customer shippable unit
- It comes in on a full pallet
- It goes out in a mixed pallet
- What happens in between?

- Take a few minutes to sketch out your answer

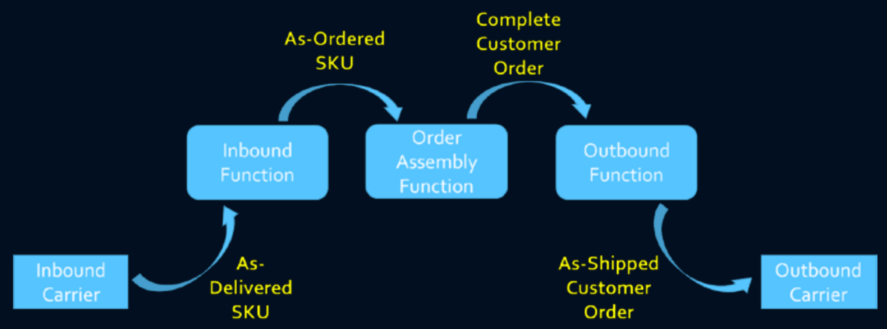
Every SKU in the FFN warehouse flow sees the following *processes*

- Unload from truck
- Receive/stage
- Move to storage location
- (pick pallet from reserve and move to forward location)
- Case/each pick from forward location
- Case/each assemble to customer order
- Customer order packed
- Customer order moved to shipping/staged
- Customer order loaded to delivery vehicle

We can generalize this specific flow to describe the generic warehouse ***functions***



For FFN



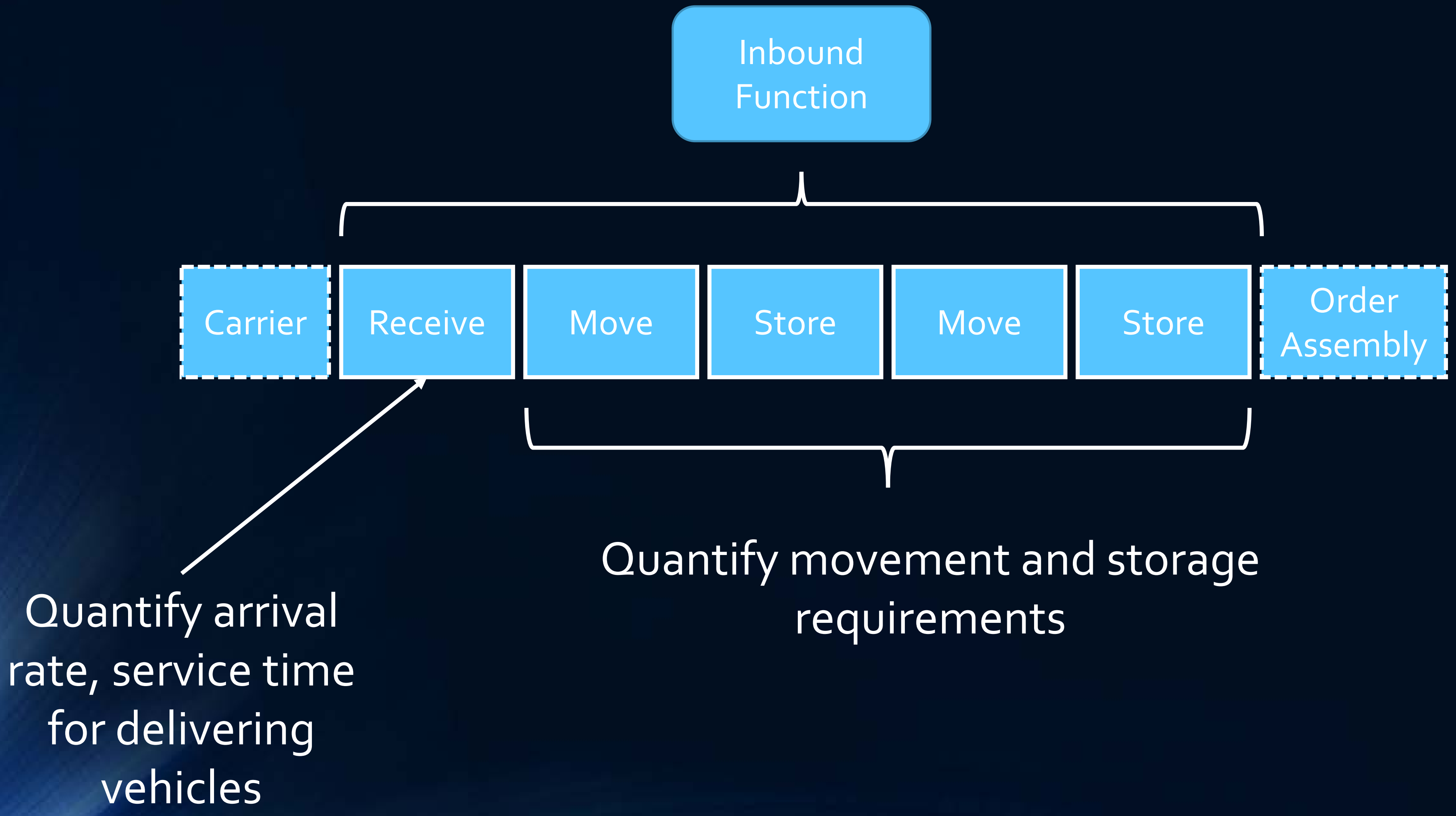
- Inbound Function "disassembles" carrier loads, receives SKUs, stages SKUs for putaway, and gets SKUs to the locations from which they will be retrieved by the Order Assembly Function
- Order Assembly is, in fact, a "manufacturing" process—it takes "order parts" from inventory and assembles them into a customer order
- Outbound Function prepares customer orders for shipment, and stages them for pickup/loading by carriers
- As-delivered and as-ordered are almost always different:
  - Pallets to cases, cases to eaches
  - What flows into Order Assembly is quite different from what flows out

Observations

- Context identified
- Flows identified
- → Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated
- Iteration

- These functions are quite well-defined, both in terms of flows and in terms of "interfaces"
- We can design each function independently, provided we have a clear definition of its interfaces *and flows*
  - But they still have to be integrated!
- The design process is almost certainly iterative
- Of course, we could try to intuit a "grand design" all at once! (Not a good idea, especially for novices…)

Elaborating the Inbound Function



Requirements for movement and storage are dictated by inbound flows

- In FFN, there are nearly 6000 active SKUs, and we can't design a different system for each one!
- We need to identify *SKU families*, with the intention that every SKU in a SKU family will be treated exactly the same way in the inbound function
- Why would we treat two SKUs exactly the same way—what makes them "similar"?
- Let's look at the FFN data…

Similarities

- Channel (direct, cross-dock, warehouse) *but…*
- Storage requirement (ambient, cool, chilled)
- Usage rate (units per week, e.g.)
- Usage per order
- Size (cube, weight)
- Resupply leadtime
- Other?
- Any consideration impacting how goods are stored or moved

SKU Profiling Analysis

- Dale Masel's unit in the CICMHE
- Textbooks and other sources
- Pivot tables in Excel
- Tableau
- Intuition!

Reality Check

- The data is !@#$, usually
- This is especially hard for students to deal with, so you need to be prepared to provide a lot of support
- Develop some general guidelines to give, when asked specific questions

- Let’s look at the FFN case data

Data "Issues"

- Mixed numbers and text (impacts Excel operations)
- Delete the "zero usage" SKUs, 38000->5700 relevant SKUs
- Nonsensical values:
  - 3 poultry SKUs account for nearly 150 million pounds shipped, roughly 3700 reefers!
- 1678 SKUs rec'd in warehouse, but only 1376 shipped…
- Total in not close to total out…
- Undocumented "units of measure" … argh!
- Etc.…
- You have to *decide* how to deal with these data anomalies, i.e., what data will you use to drive your design?

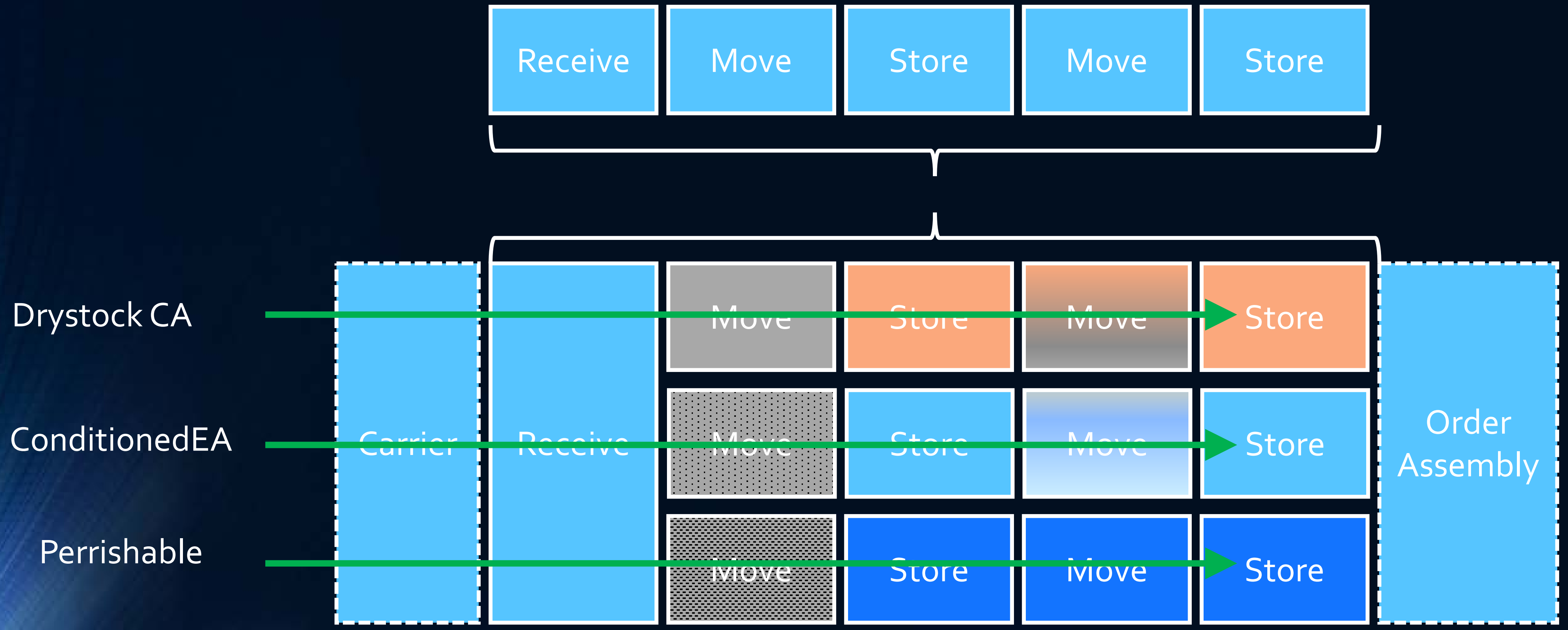
Input Families

Row Labels	D	L	M	P	Grand Total
WH					
CA					
Sum of Total Usage	3831617	202793	5271	2348955	6388636
Count of SkuNbr	671	86	6	202	965
Sum of Tot Cube	3916112	154336	4346	1766554	5841347
EA					
Sum of Total Usage	1727484	1452443		43588	3223515
Count of SkuNbr	36	343		2	381
Sum of Tot Cube	106739	748182		9153	864075
LB					
Sum of Total Usage				7906336	7906336
Count of SkuNbr				28	28
Sum of Tot Cube				4010209	4010209
PL					
Sum of Total Usage	1144				1144
Count of SkuNbr	1				1
Sum of Tot Cube	54912				54912
WH Sum of Total Usage	**5560245**	**1655236**	**5271**	**10298879**	**17519631**
WH Count of SkuNbr	**708**	**429**	**6**	**232**	**1375**
WH Sum of Tot Cube	**4077763**	**902518**	**4346**	**5785916**	**10770543**

Input Families:

- DrystockCase: 671 SKUs; e.g., pallet rack, forward at floor level
- ConditionedEach: 473 SKUs (36 Drystock/EA, 92 Alcohol/Case and 345 EA-Alcohol or Choc/Tea; e.g., pallet rack, forward at floor level
- Perishable: 232 SKUs, all P, 202 CA, 2 EA, 28 LB; e.g., pallet rack forward at floor level

Elaborating the Inbound Function



List of "functions":
Receive
6 Moves
3 Stores

NOTE:
I can come back later and change the families, or change the functions or change how I group functions.

What we need to specify for each function

- For each Move function:
  - The number or rate of move transactions
  - The properties of the move transactions (weight, volume, etc)
- For each Store function:
  - The number or rate of store/retrieve transactions
  - The storage capacity
  - The properties of the storage (constraints to be satisfied)

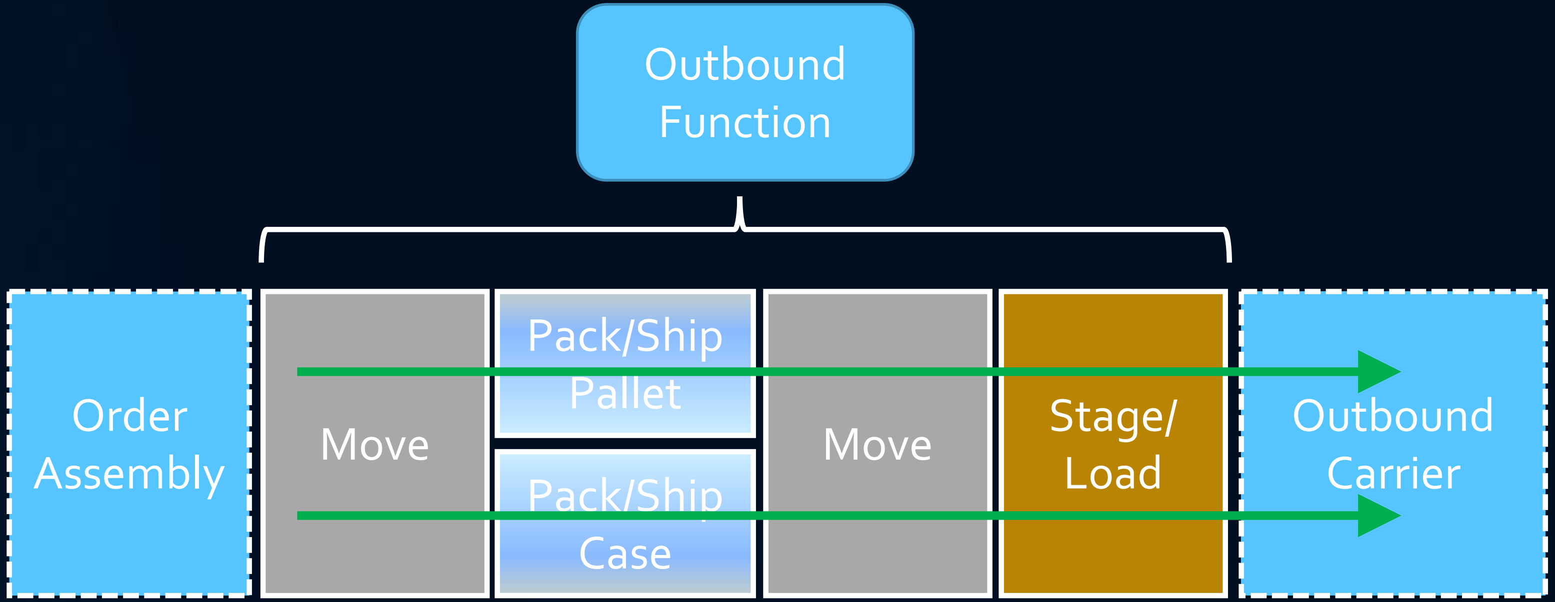
Data Limitations

- We only have totals by month, no information about actual received shipments.
- So, the design team has to make some reasonable assumptions
- *In this case, what might be some reasonable assumptions?*

Elaborating the Outbound Function

We don't have much to go on here…
Assume orders take two forms: full/partial pallet, or one or more cases. Assume that both kinds of orders are moved in the same way from Order Ass'y to Pack/Ship, and then to Stage/Load

Four functions:
Move
Pack/Ship Pallet
Pack/Ship Case
Stage/Load



Unfortunately, for this case problem, we don't have detailed information about daily shipments, so we'll have to make some assumptions. *What kinds of assumptions would be reasonable, given the overall scenario?*

What we need to specify for each function

- For each Move function:
  - The number or rate of move transactions
  - The properties of the move transactions (weight, volume, etc)
- For Pack/Ship:
  - The rate at which orders are packed/shipped and the characteristics of orders (needed to estimate resource requirements)
- For Stage/Load
  - The arrival rate of orders to load, the arrival rate of carrier vehicles (queuing related properties)

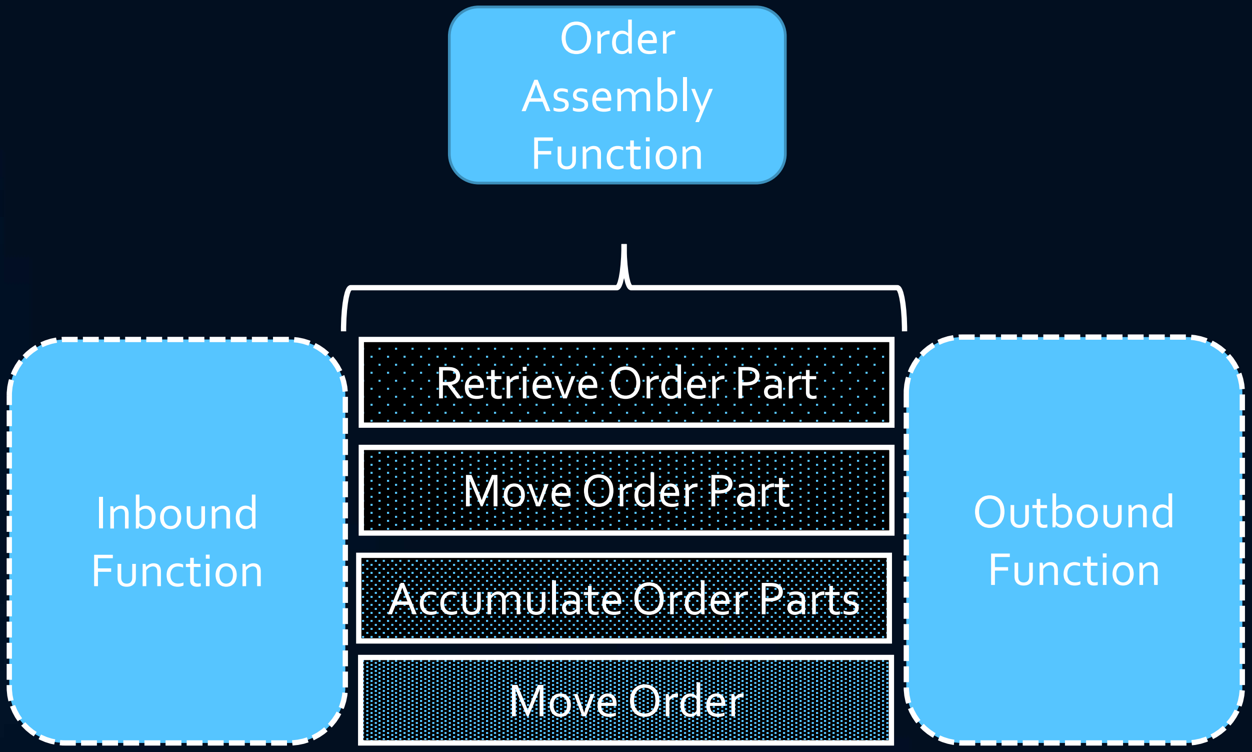
Data Limitations

- We only have totals by month, no information about actual customer shipments.
- So, the design team has to make some reasonable assumptions
- *In this case, what might be some reasonable assumptions?*

Some Really Important Observations

- The InBound and OutBound functions are “linear flows”
- The implementation of these functions is, more or less, independent of the “control” decisions involved, such as putaway location or dock door assignment
- We can at least think about, and often specify, the resources needed to embody these functions first, and then worry about the associated control systems later
- ***This makes InBound and OutBound quite different from Order Assembly!***

Elaborating the Order Assembly Function
(this is where all the magic occurs!)



Order assembly does not decompose nicely into a linear functional flow. It's a system, and has to be thought of that way.

What can we infer from the data?

- Rates of “order parts” retrieval
  - Eaches, cases, pallet retrieval rates
- Rates of accumulation
  - Relative frequencies of lines per order
  - Accumulation composition
    - Mix of each, case, pallet
- Rates of “move” really depends on the specific system selected

FFN Order Assembly Data

- Don't have actual customer order data
- Have to make some assumptions

- *What assumptions seem reasonable?*

"Process Check"

- Take-aways
  - Semantics
  - Identify flows
  - Identify functions
  - Identify flow families
- Next
  - Architecture

System "Requirements" Decision (Identify flows, Identify functions, Identify flow families)

System Design Decision (Architecture)

- Context identified
- Flows identified
- Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated
- Iteration

Architecture

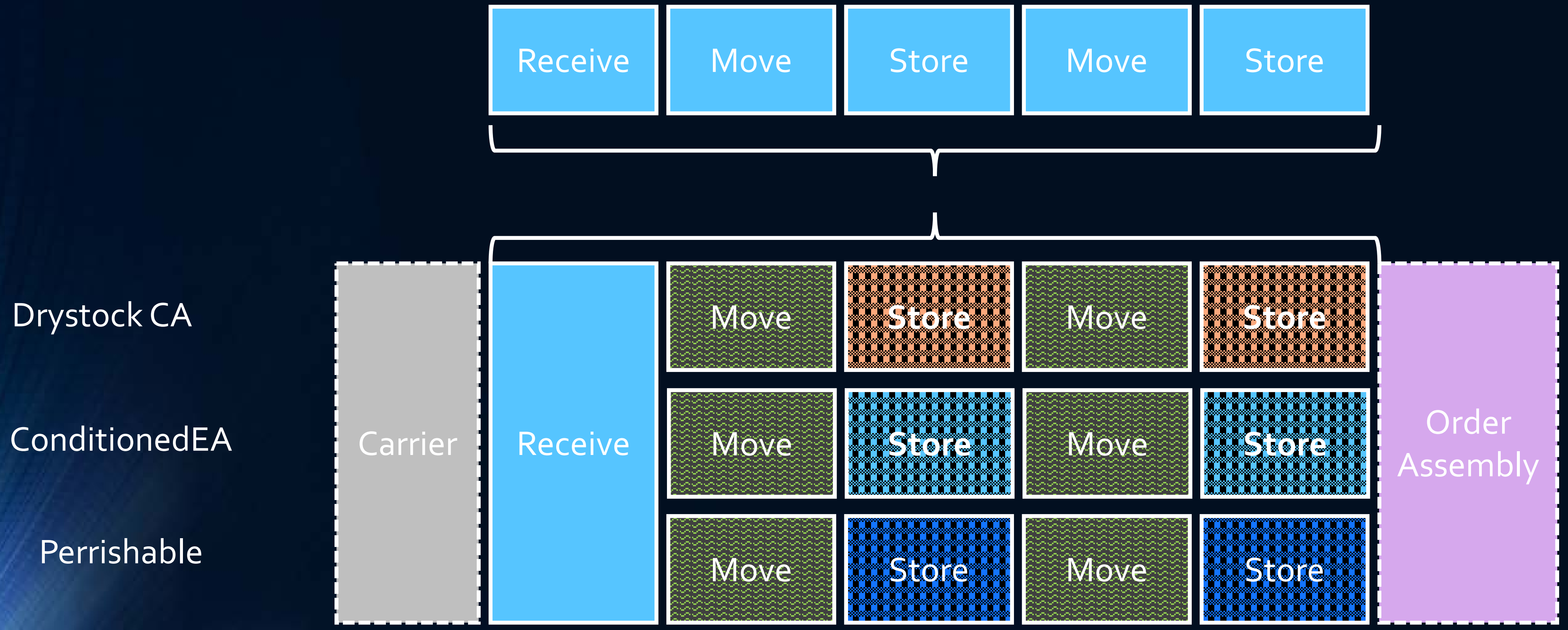
- Basic idea is to aggregate functions into subsystems.
- Examples:
  - Architecture of an automobile: frame, body, drive train, cabin
  - Architecture of an airplane: wing, fuselage, propulsion
- Motivation: break the design problem up into "homogeneous" pieces that can be worked on independently, but integrated

How do you make these “architectural” decisions?

- You have to know that the functions can, in fact, be performed by a single system.
  - In an automobile, you wouldn’t combine a “provide rotational power” and “provide communication access” functions
- You probably use some prior knowledge of the kinds of technologies that can be used for the individual functions
- I.e., it’s not a random assignment!

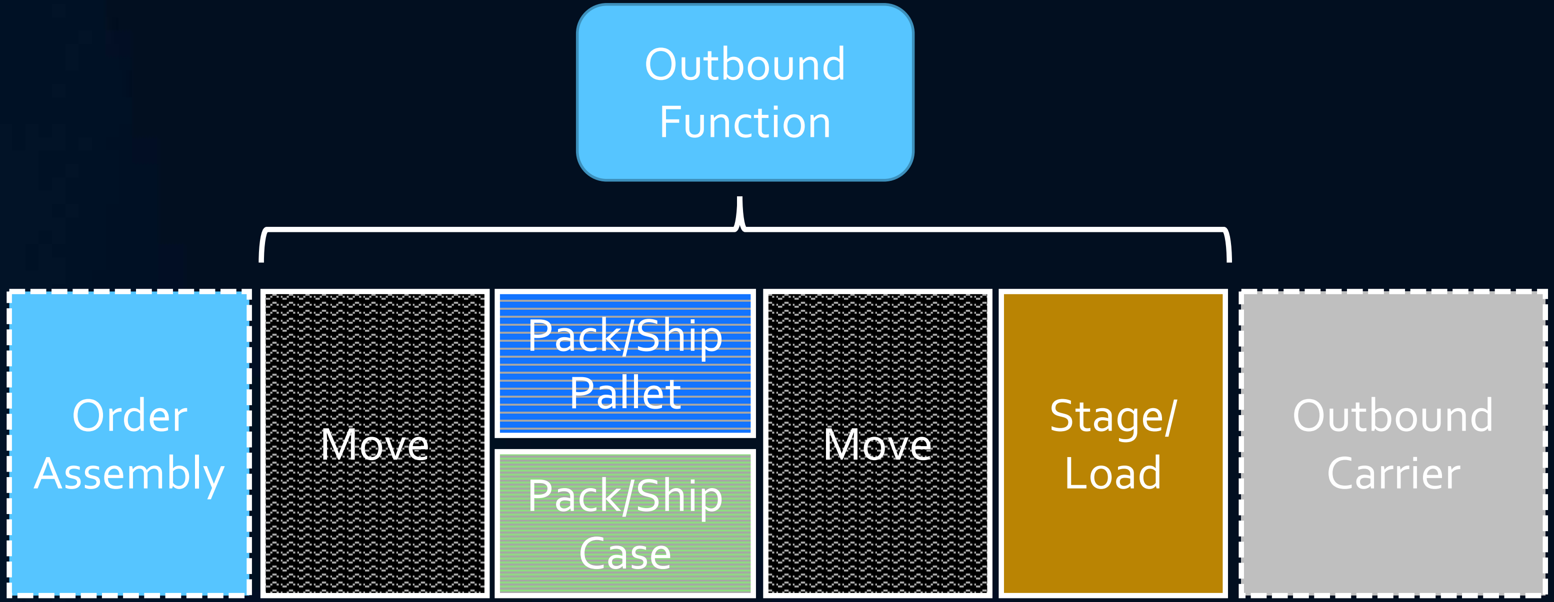
An opportunity to introduce material handling equipment and systems…

Inbound Functional Architecture



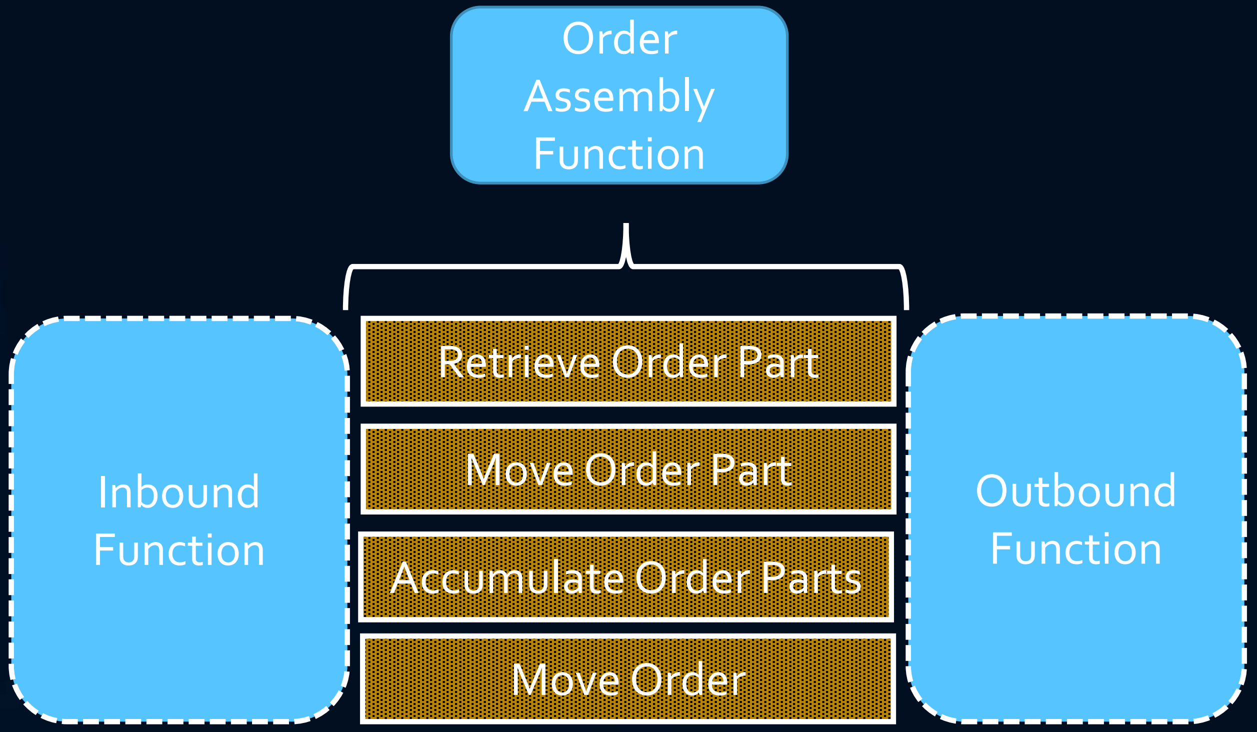
All move functions combined into one system (maybe lift trucks of some sort).
For each flow family, a single store system for both reserve and forward/pick
Five subsystems to design: receiving, moving, and three storage systems.

Outbound Functional Architecture



One move subsystem, two pack/ship subsystems, and one stage/load subsystem

Order Assembly Functional Architecture



Order assembly does not decompose nicely into a linear functional flow. It's a system, and has to be thought of that way—one system that performs all four functions.

Selecting Technologies

- Context identified
- Flows identified
- Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated
- Iteration

- Each subsystem has well-defined input and output requirements
- Each potential technology has well-defined capabilities (processes that it can execute)
- Selecting technology is a matter of matching requirements and capabilities
- There can be many feasible combinations, and evaluating each one can be time consuming, because you have to size and configure the technology in order to assess its performance and cost
- You want some guidelines for identifying "promising" technologies

So, how does the designer choose?

- Use knowledge and judgement to reduce the set of candidates.
- There are many resources that can be useful, see .e.g., http://www.mhi.org/downloads/learning/cicmhe/colloquium/2010/apple.pdf and many other sources accessible through google

Suppose the technology selections are:

- Inbound
  - Lift trucks, pallet rack
- Order Assembly
  - Single order pick to pallet using lift trucks
- Outbound
  - Lift truck, accumulate case orders to move to Stage/Load

Note: there some pretty obvious variations of these decisions which should also be examined…

Each subsystem has to be sized and configured

- Inbound
  - Number of dock doors, space for staging
  - Storage:
    - Capacity from inventory analysis
    - Tiers, aisles
  - Lift trucks:
    - Number based on move rate and distance travelled
      - How to incorporate reserve-to-forward moves?
      - Queuing type analyses

Here, you can introduce many different kinds of analyses to support the sizing and configuring

Each subsystem has to be sized and configured

- Order Assembly
  - Number of order pickers
    - Depends a great deal upon
      - Where SKUs are located in storage
      - How pickers are routed, e.g., to minimize in-the-aisle conflicts
- Outbound
  - Number of pack/ship stations and staging for each
  - Size of Stage/Load area

Here, you can introduce many different kinds of analyses to support the sizing and configuring

Sizing and configuring is mostly analysis, not decision-making!

There are many, many sources of analysis models that can be used.

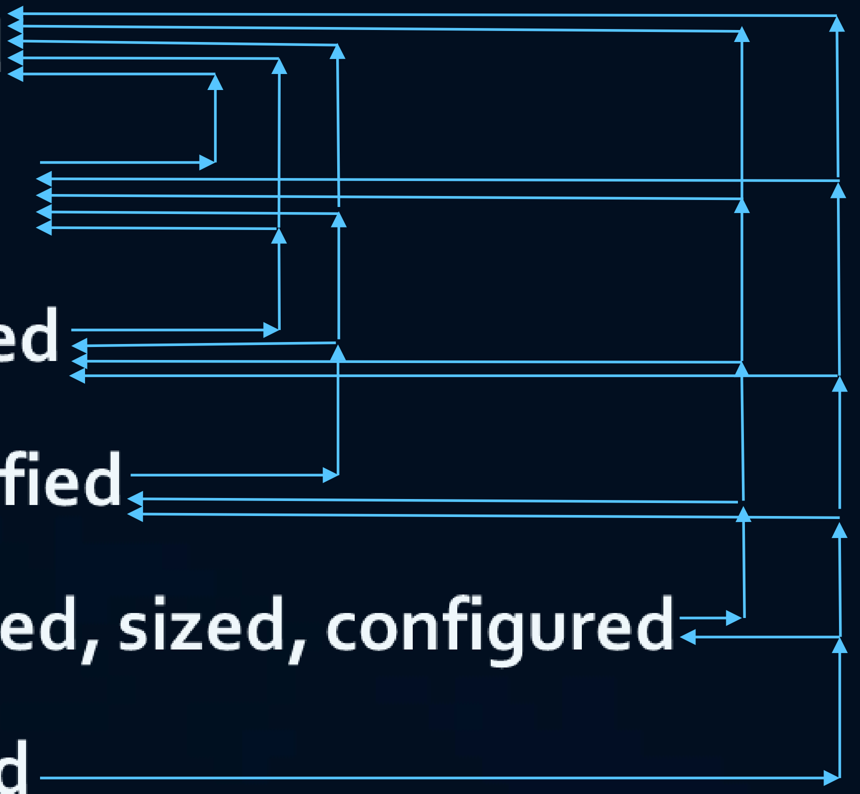
System Evaluation

- Context identified
- Flows identified
- Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated
- Iteration

- Evaluating the system may not be as simple as "adding up" the evaluations of each subsystem
  - How do the subsystems interact?
  - Do the interactions have any impact on performance?
- May do some "tweaking" of the design to improve the system evaluation.

Iteration in System Design

- Context identified
- Flows identified
- Functions identified
- Architecture specified
- Technology selected, sized, configured
- Solution evaluated



At any point in the process, the designer can go back to change a previous decision!

Analysis to Support Design Decision Making

- Inventory modeling to assess the storage capacity required
- Analysis models to configure, e.g., for a given “shape factor”
- Once a storage system has been “sized” and “configured” and then analysis can estimate the travel time for putaway and for replenishing forward pick locations.
- For many picker-to-goods systems, there are analytic models of picker performance
- Of course, simulation …

Principles

- Deciding how to deal with the data is a design decision!
- Specifying SKU and Order families, functions, architecture, and selecting technologies are design decisions
- Sizing and configuring is a design decision, but it often can be done algorithmically—optimizing some performance attribute
- SKUs are inventory as long as they are in the InBound Function
- SKUs in Order Assembly are order parts

CICMHE design competition cases are a great way to teach design principles, and students generally are much more engaged with design cases than with traditional lecture.

Just do it!

To the MHTI 2017 participants, to David Porter, to CICMHE, and to the many colleagues who have influenced my thinking about design,

Thank you!