System Architecture
IAP Lecture 5

Ed Crawley

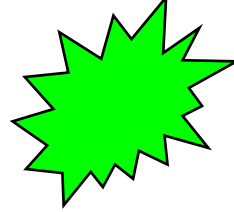
Jan 26, 2007

Rev 2.0

Today's Topics

- **Reflection on Operations, Interfaces, etc.**
- **Worked example - TCP**
- **Alternative representations of Process-Object architecture**
- **Upstream influences - Beneficiaries, Needs and Goals**
- **Worked example - ServeCo**

Reflections on Operations, Interfaces, etc?



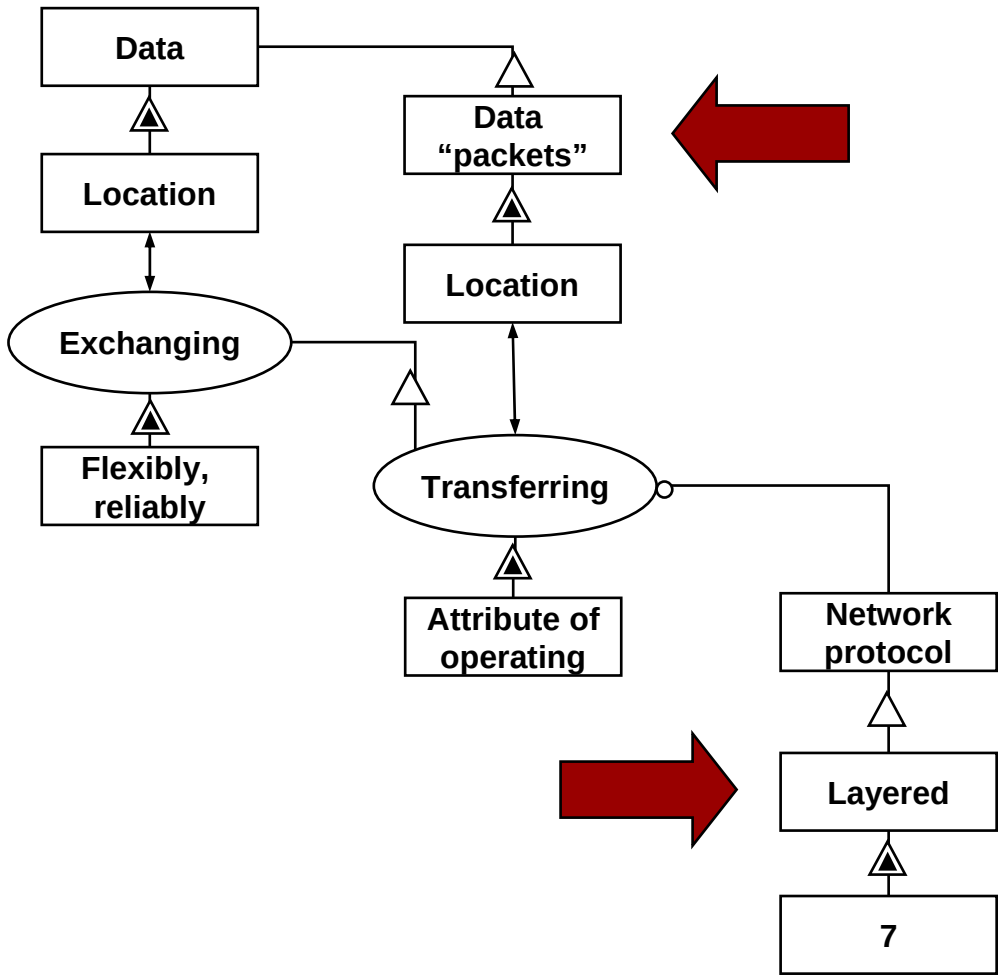
- **What are the elements the contribute to:**
  - **Other value processes?**
  - **Supporting processes?**
  - **Interfacing processes?**
- **Suggest a level 1 modularization? Why?**
- **What are important operational sequence?**
- **What is an important interface?**

Concepts and Architectures in Information Systems

- **In information and software enabled systems, concepts and their development into architectures are captured in different ways at various levels**
- **Low level are *algorithms* and their implementation, e.g. bubblesort**
- **Application domain software are *patterns*, e.g. bridge**
- **Higher level application domain software is more classical allocation of functionality to modules and procedures, and definition of interfaces**
- **In network software, concepts and architectures are captured in protocols, such as Transport Command Protocol (TCP), and at a higher level, the entire architecture of the OSI seven layer model**

Solution Neutral Function and Specific Concept - Network

- Data exchanging is the solution neutral function
- There are very many ways to do this, an open layered network protocol being just one



Layers of Form, and Associated Function (?)

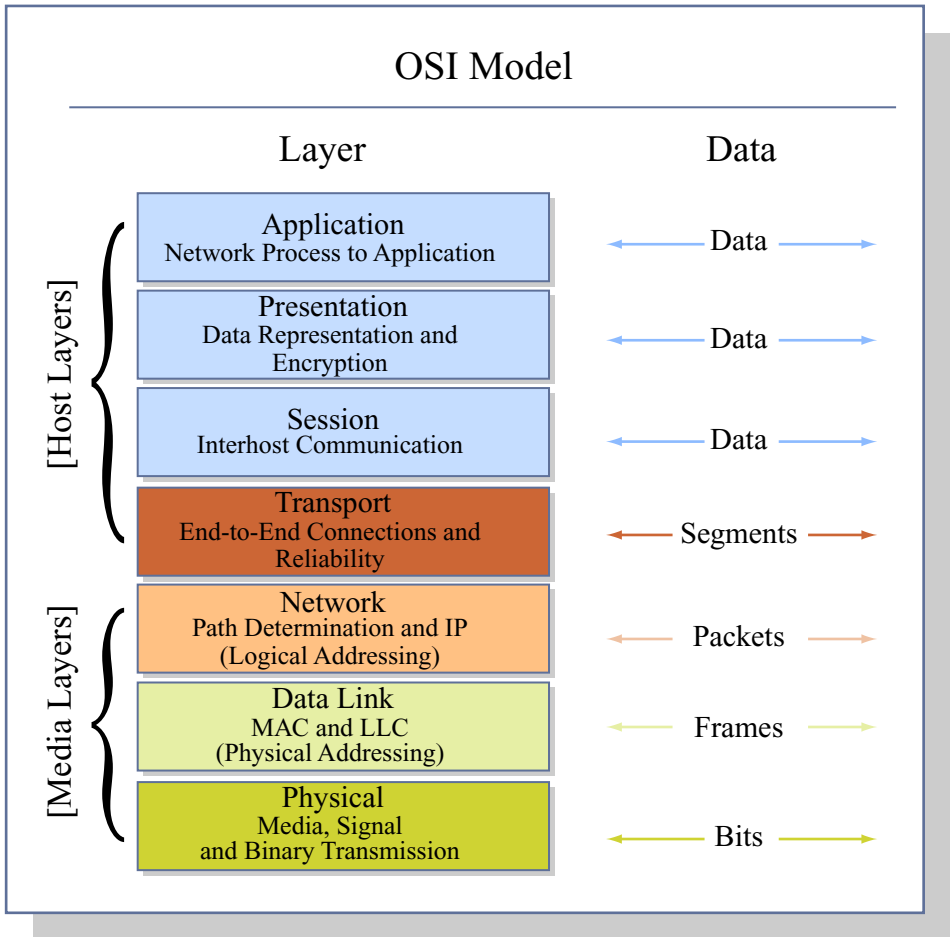


Figure by MIT OCW.

http://www.3mfuture.com/network_security/arp-guard-arp-spoofing.htm

Layer Function

- **For each layer, identify the operand, data related process and control related process**

	Operand	Data process	Control process
Application			
Presentation			
Session			
Transport			
Network			
Data			
Physical			

Assigning a Function to a Layer

- **To which layer are we going to assign the function of "reliable" transmission:**
- **The physical layer, to make sure bits are exchanged?**
- **The network or transport layer, to make sure that segments/packets are exchanged?**
- **The application layer, to make sure data sets were exchanged?**

- **All?**
- **None?**

Whole Product, Use Context and Implied Interfaces

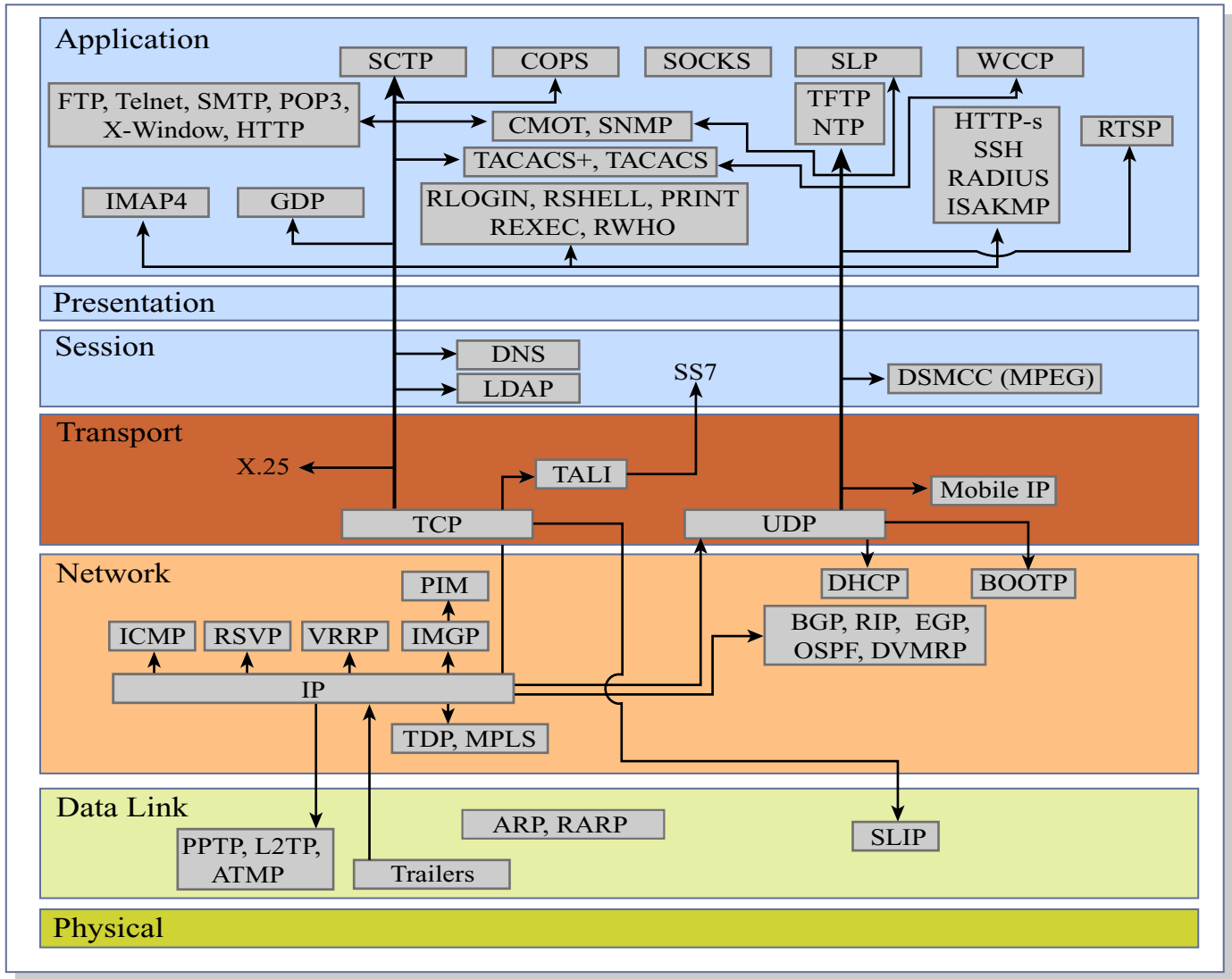


Figure by MIT OCW.

http://www.protocols.com/pbook/tcpip1.htm

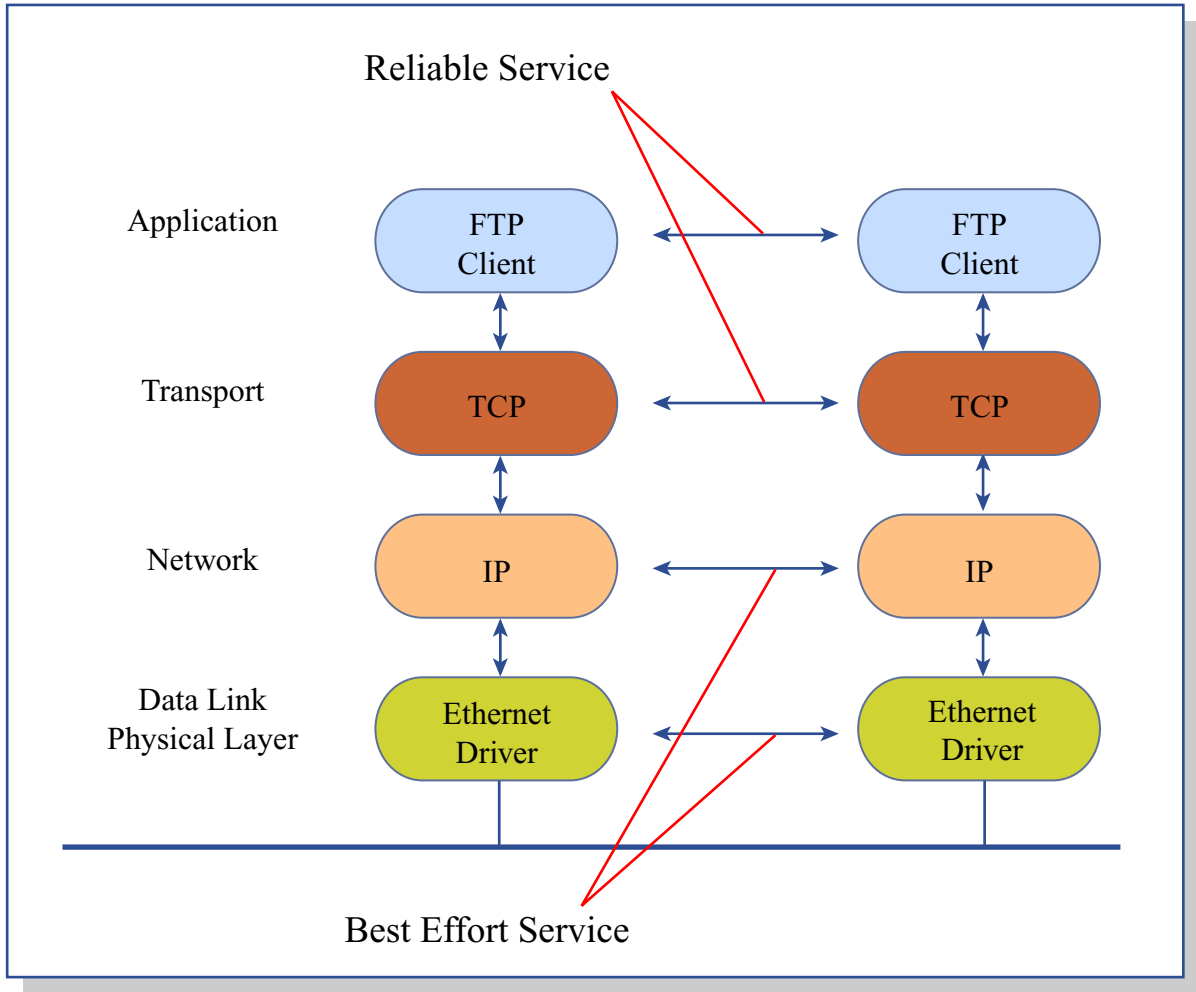


Figure by MIT OCW.

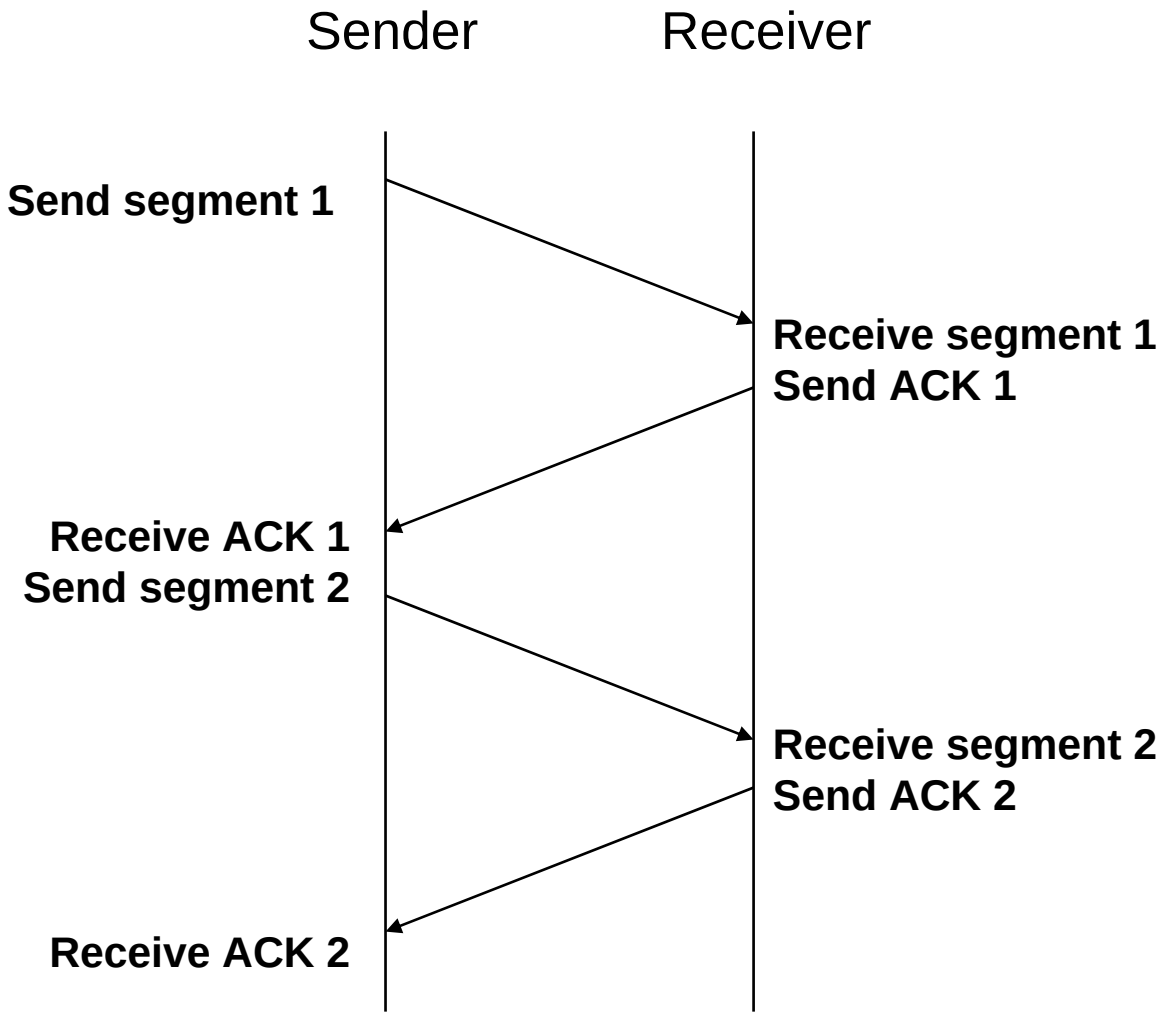
Why Layers?

- **Layered architectures have great advantages**
  - **Flexibility in application**
  - **Flexibility of dynamic connections from one layer to another**
  - **Ability to provide alternatives in assignment**
  - **Isolation of changes in one layer from others**
- **Layered architectures have disadvantages as well**
  - **Inefficiency due to overhead of adding information at each layer**
  - **Interface inefficiency**
  - **Inefficiency due to the fact that sometimes one needs to reallocate function across layers or combine**

In general, there is always a tension between flexibility and optimality in architecture

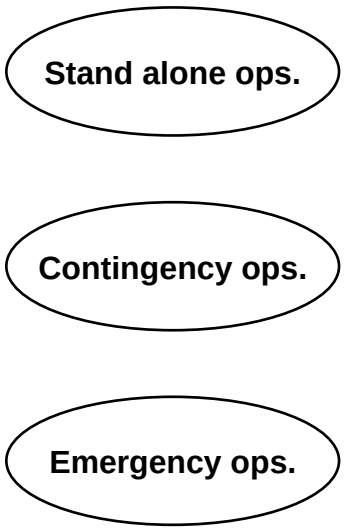
TCP Primary Operating Sequence

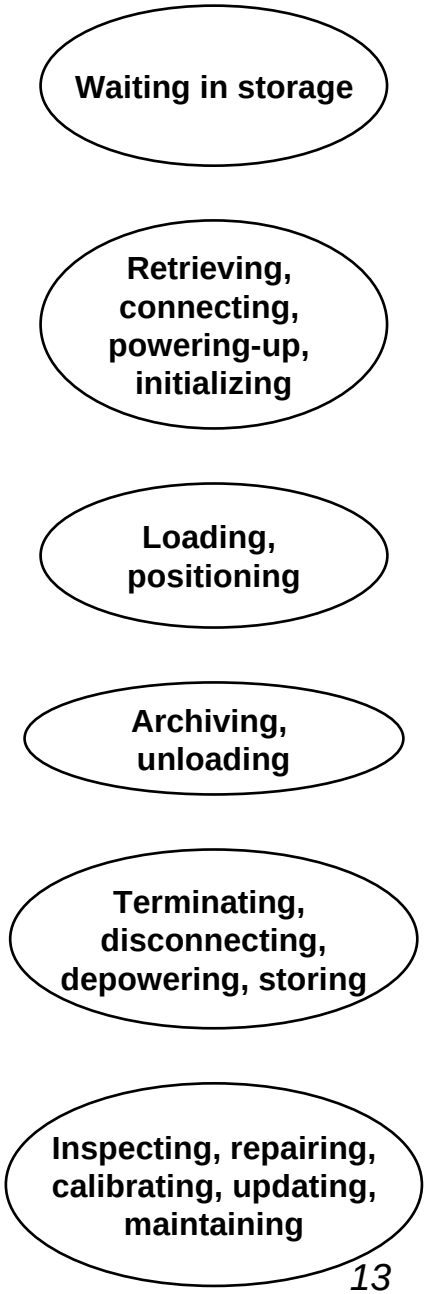
- **Sender sends a segment, with a number**
- **Receiver receives and acknowledges that segment**



Fuller Operating Sequence?

- Stand alone?
- Contingency?
- Emergency?
- Commissioning?
- Decommissioning?
- Maintaining?
- Is real clock time important?





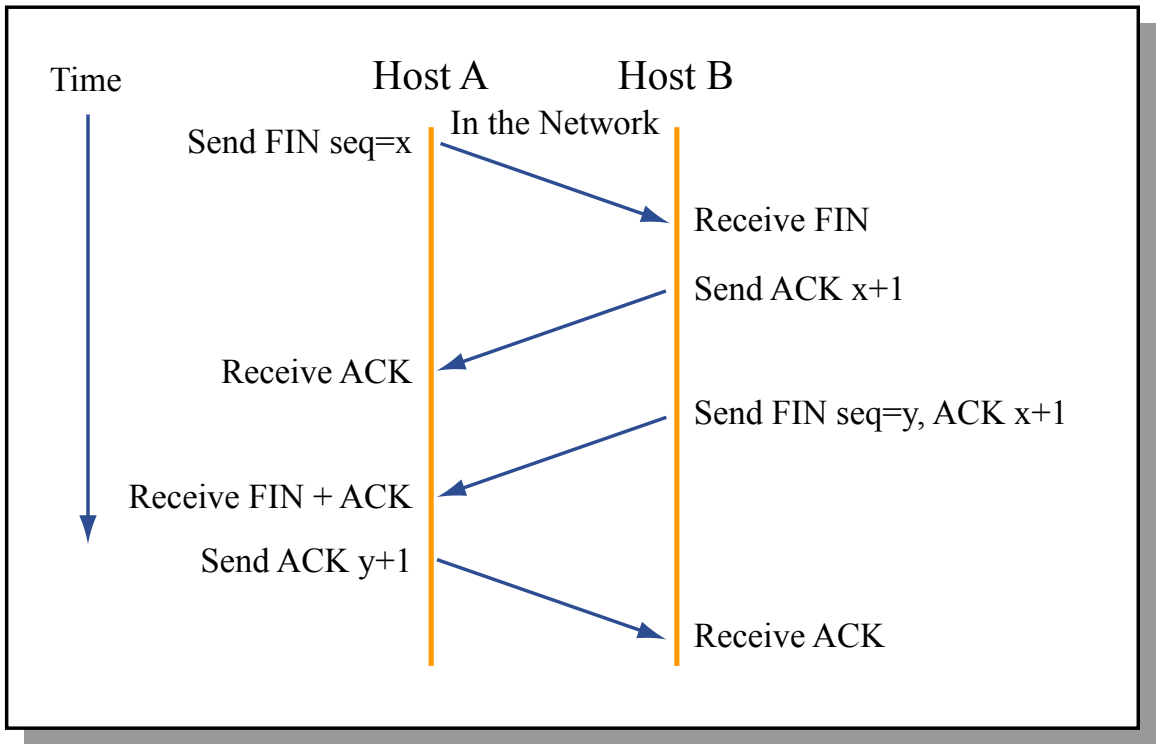


Figure by MIT OCW.

TCP Connection Termination

TCP Header

RFC 793 – Transmission Control Protocol

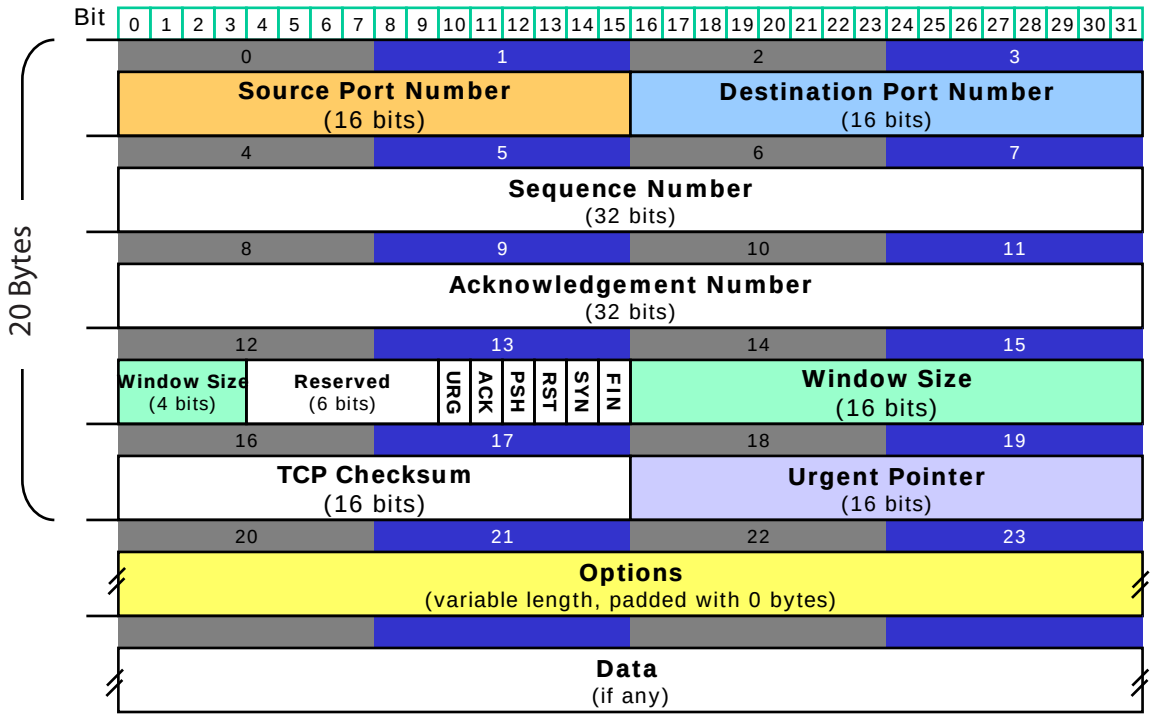
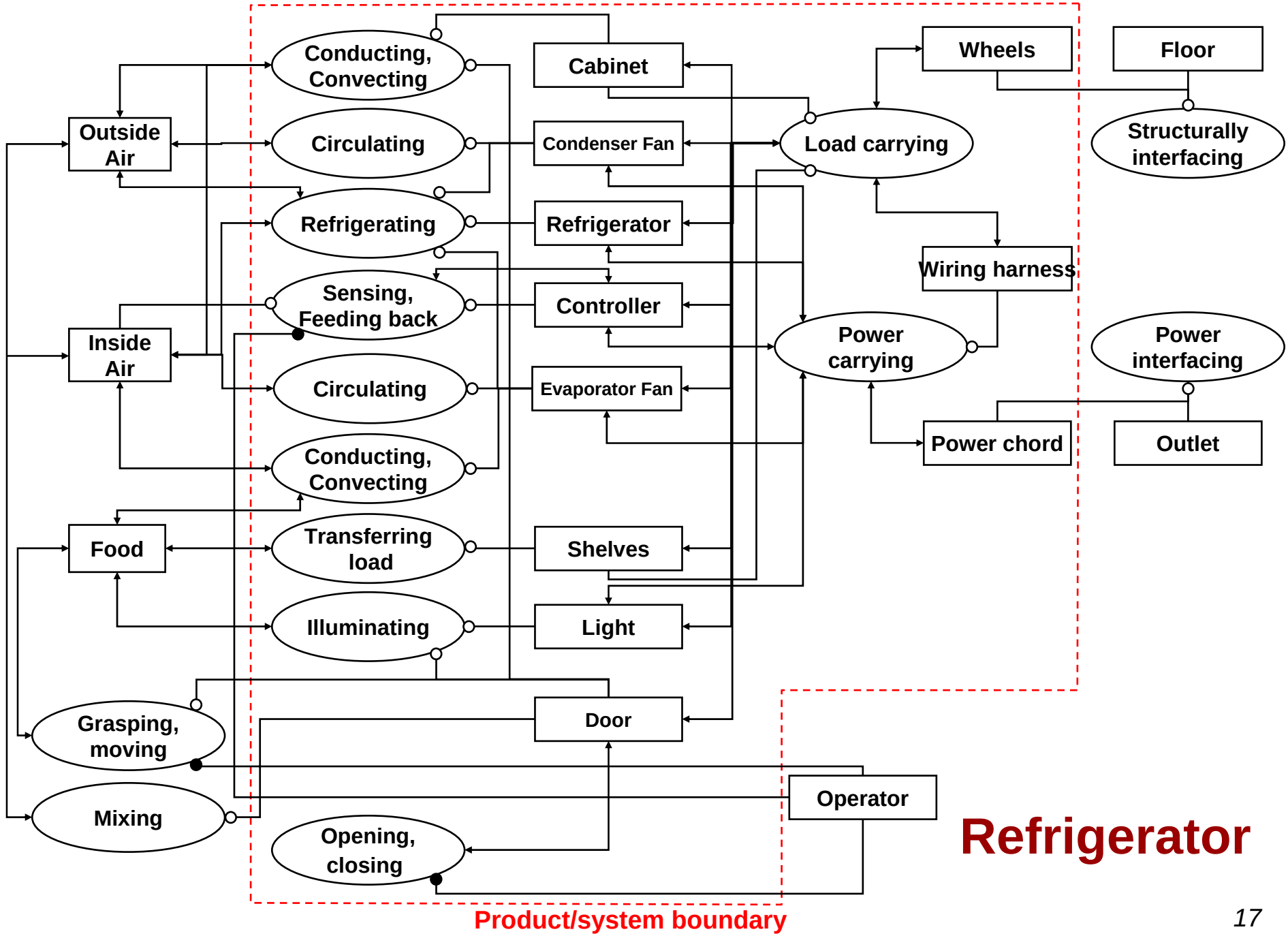


Figure by MIT OCW.

Representing Object-Process Architectures

- There are various ways to represent objects and processes in a model:
  - Explicitly show objects and processes (e.g. OPM)
  - Explicitly show processes and suppress objects
  - Explicitly show objects and suppressed processes
  - Show only objects
- Each has its advantages and disadvantages
- Sometimes use links of various classes to label some additional information



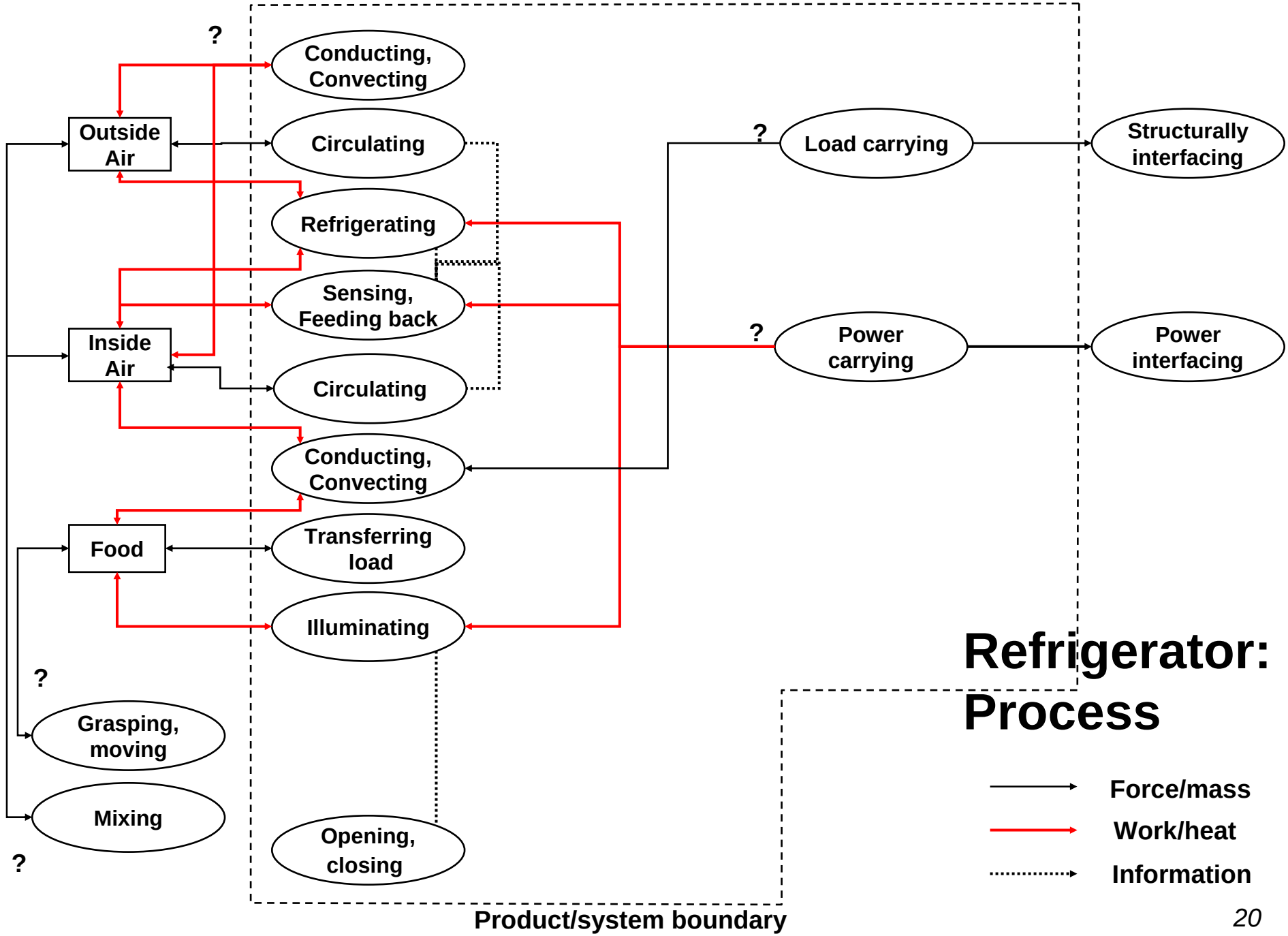
Alternative Model - Processes Model

- Alternatives to OPM exist, but tend to hide or suppress information
- A process based model:
  - Shows the processes, and objects which are inputs and outputs
  - Hides the objects which are instruments
  - Makes it difficult to identify the processes that take place among the instrument objects (the supporting processes), and the operational processes
  - Often used with different connector types to indicate the type of object as output, but this is often ambiguous
- Good for identifying the processes on the value chain

Classes of Process Links

Process links fall into classes:

Matter	Mechanical	Mass exchange	Passes flow to
		Force/momentum	Pushes on
	Biochemical	Chemical	Reacts with
		Biological	Replicates
Energy		Work	Carries electricity
		Thermal energy	Heats
Information	Signal	Data	Transfers file
		Commands	Triggers
	Thought	Cognitive thought	Exchange ideas
		Affective thought	Impart beliefs



Alternative Model - Object/Suppressed Process Model

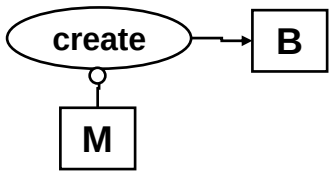
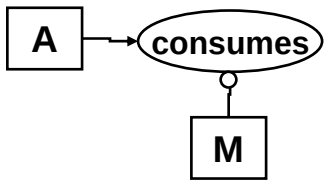
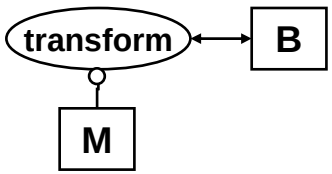
- **An object based model**
  - **Shows inputs, outputs and instrument objects**
  - **Represents the processes by arrows that pass among the objects**
  - **The arrows represent the operating process topology of the system**
  - **Lables on the arrows indicates the process and the states of the objects influenced**
  - **Can capture all or almost all the information in an OPM, but the importance of the processes, and the ability to think in process domain are reduced**
- **Good for knowledge capture and communicating with those who can't read OPM's**

Translation OPM to Object/SP Model

- With care, you can translate most or all of the information on an OPM to an Object Model
- Following charts show representative interactions on an OPM, and how they would appear on an Object Model
- There are some interactions that are difficult to capture
  - Conditional instruments
- There are actually a few things that have greater clarity in an Object Model
- If the model suppresses operational processes, it becomes the sought after operational structural model

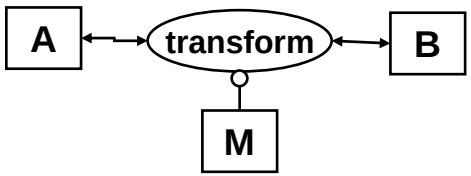
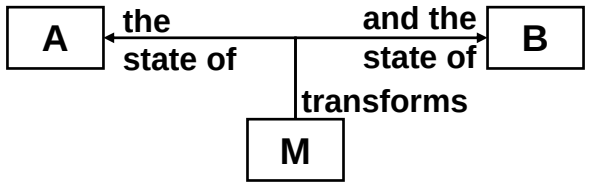
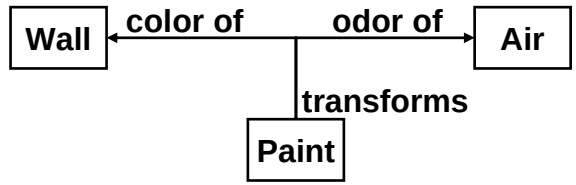
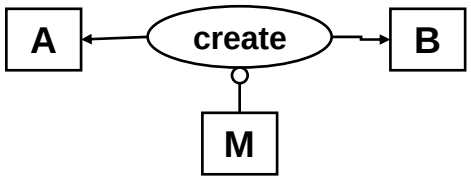
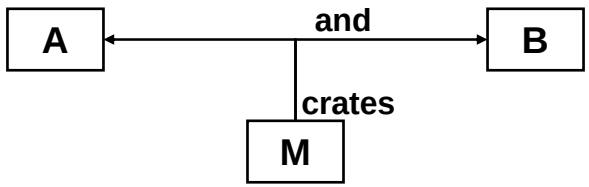
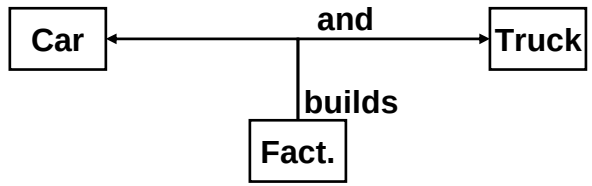
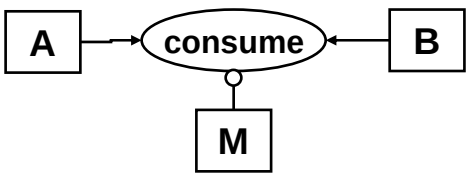
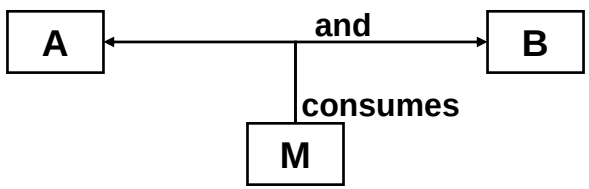
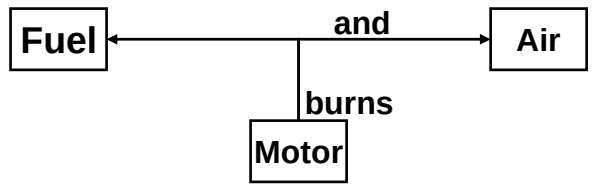
Suppressed Processes

Single Operand

OPM	Template	Example
	M creates B	Fact. builds Car
	A is consumed by M M consumes A	Part is used up by Fact. Fact. uses up Part
	M transforms the state of A	Paint transforms the color of Wall

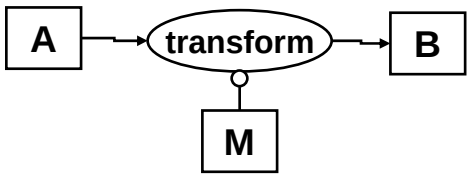
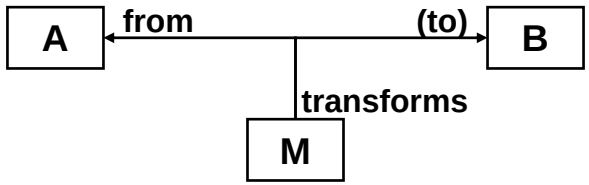
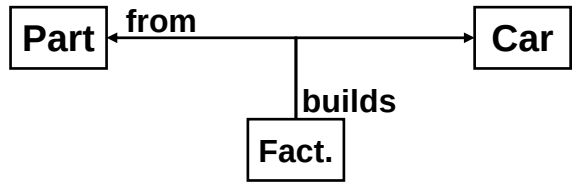
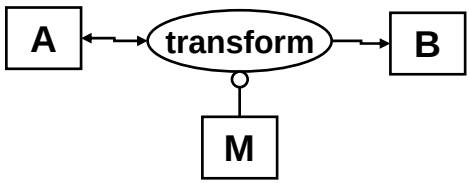
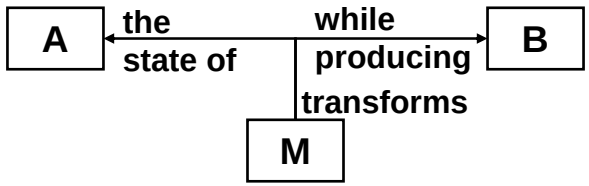
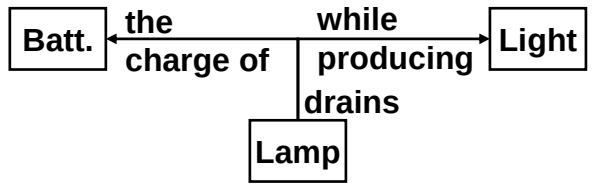
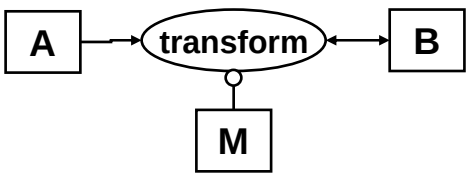
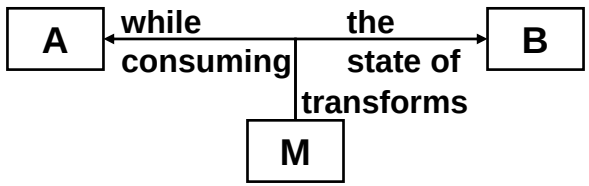
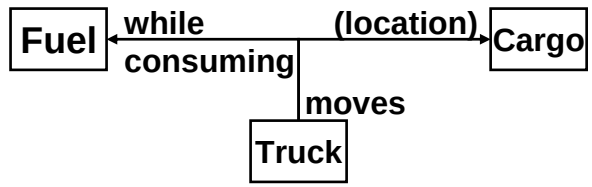
Suppressed Processes

Two Operands, Same Process

OPM	Template	Example
		
		
		

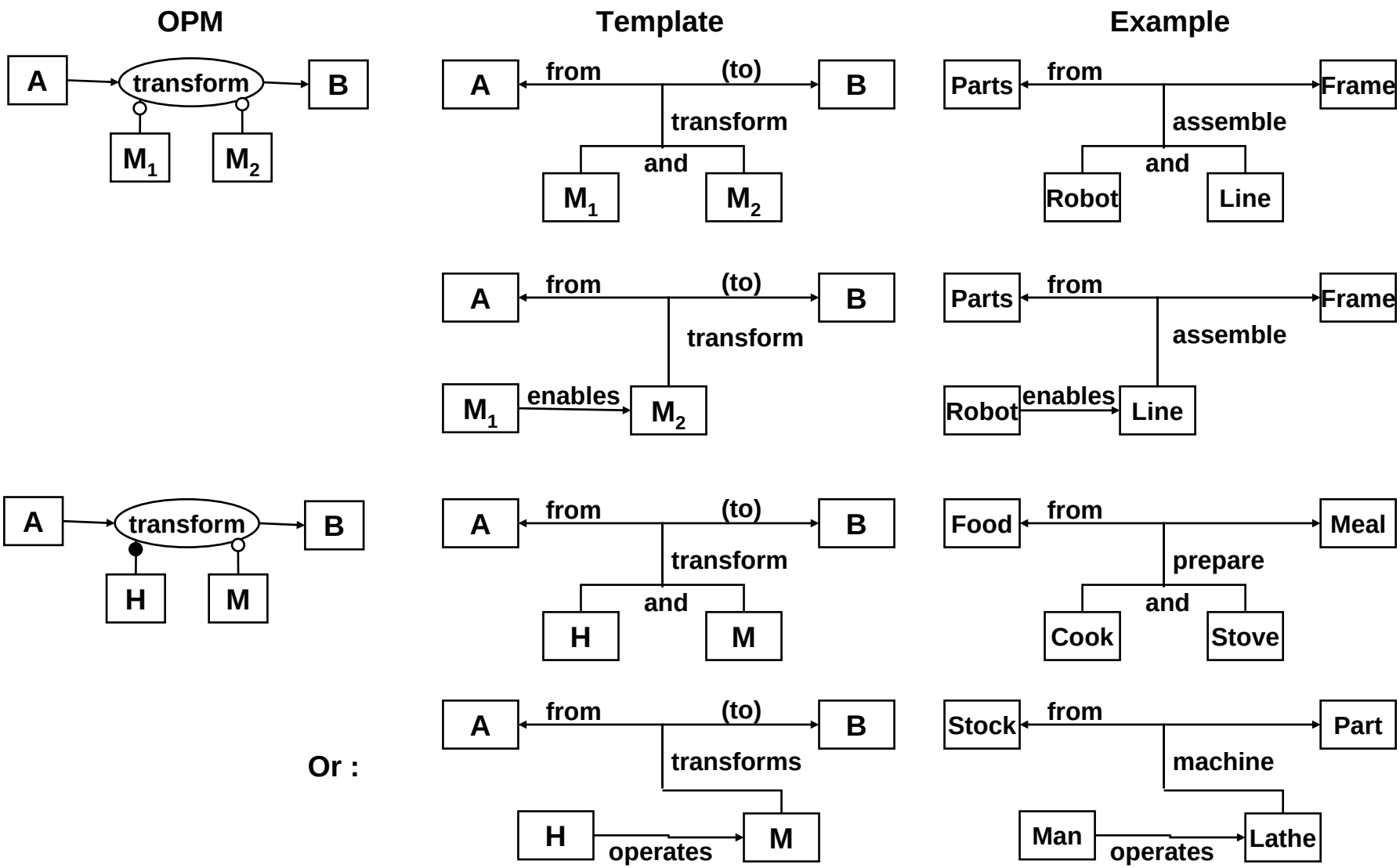
Suppressed Processes

Two Operands, Different Processes

OPM	Template	Example
		
		
		

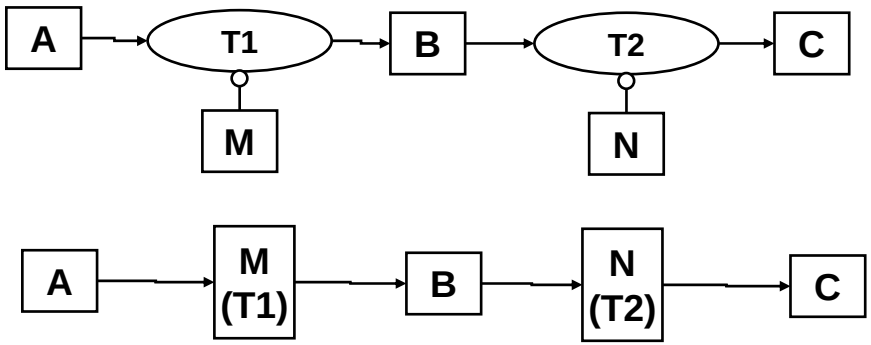
Suppressed Processes

Two Instruments

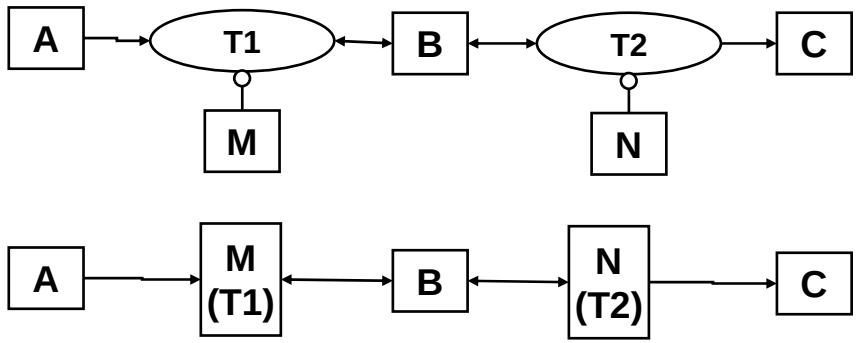


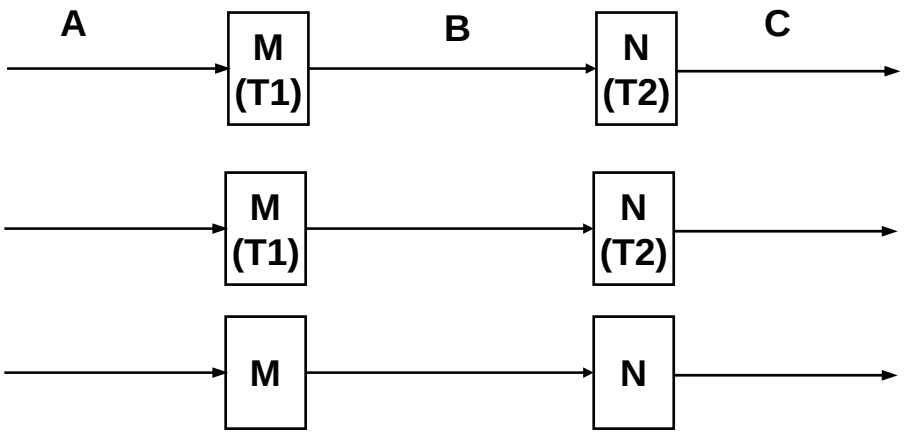
Transforming Process Object Models to Objects Only

Simple input-output



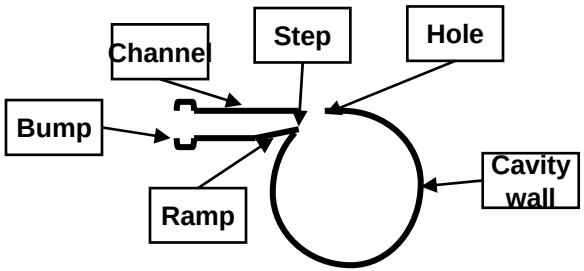
More common affecting





Next is nonsense

OPM Whistle Architecture



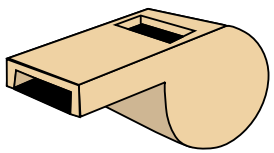
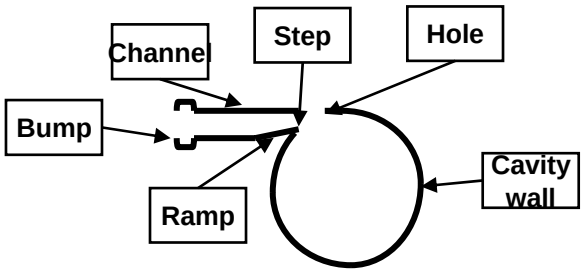


Figure by MIT OCW.

Creating
Operator
Making tone
Whistle
Aligning/ Transporting
Bump
Flow
Deflecting/ Accelerating
Channel
Ramp
Venting
Step
Vortex
Creating
Hole
Exciting
Cavity wall
Tone (internal)
Resonating/ amplifying
Star
Ring
Tone (radiated)
Coupling
External Air
Product/system boundary

Object Suppressed Internal Processes - Whistle



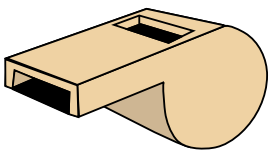
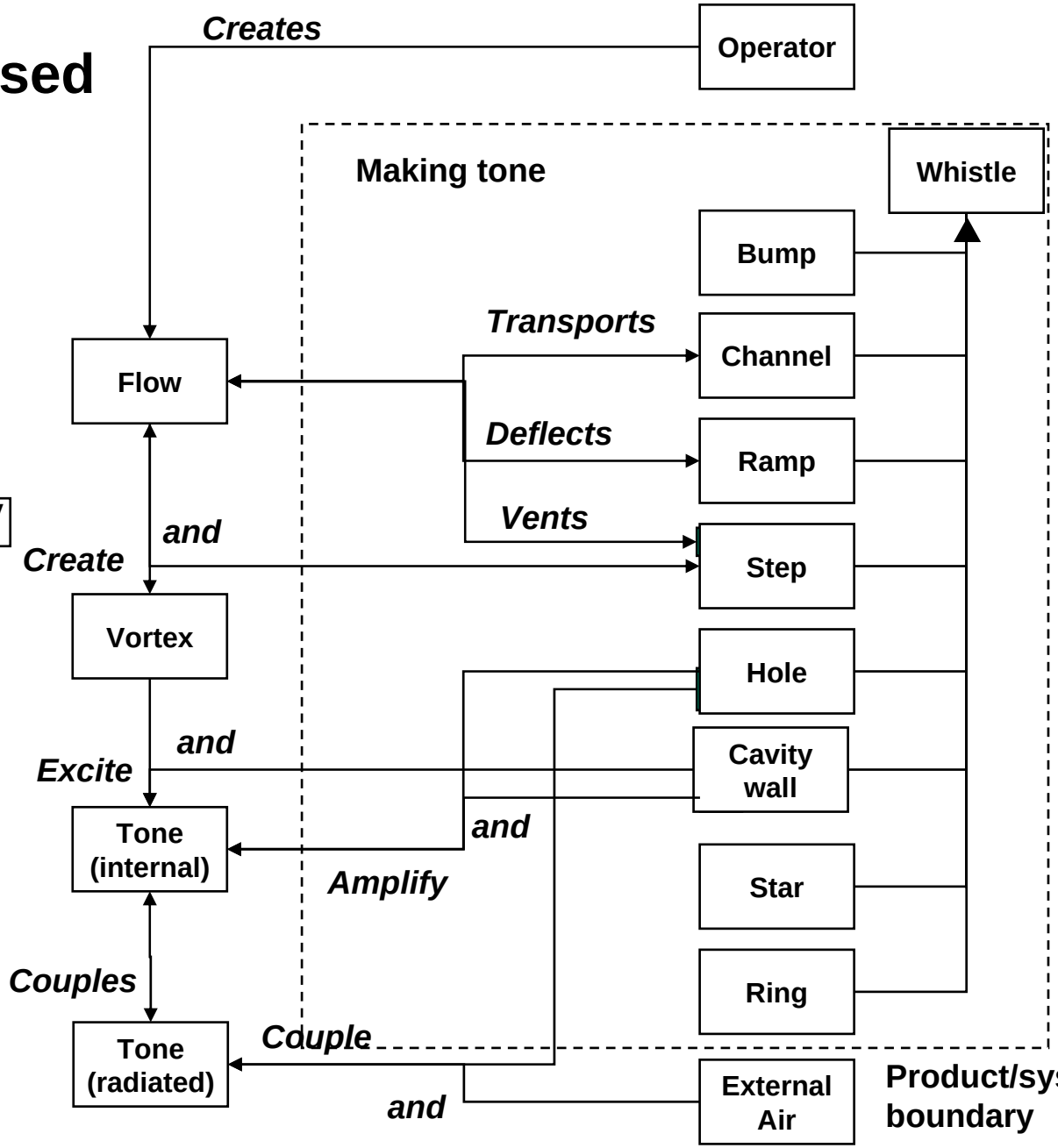


Figure by MIT OCW.



Process-Object Architectures

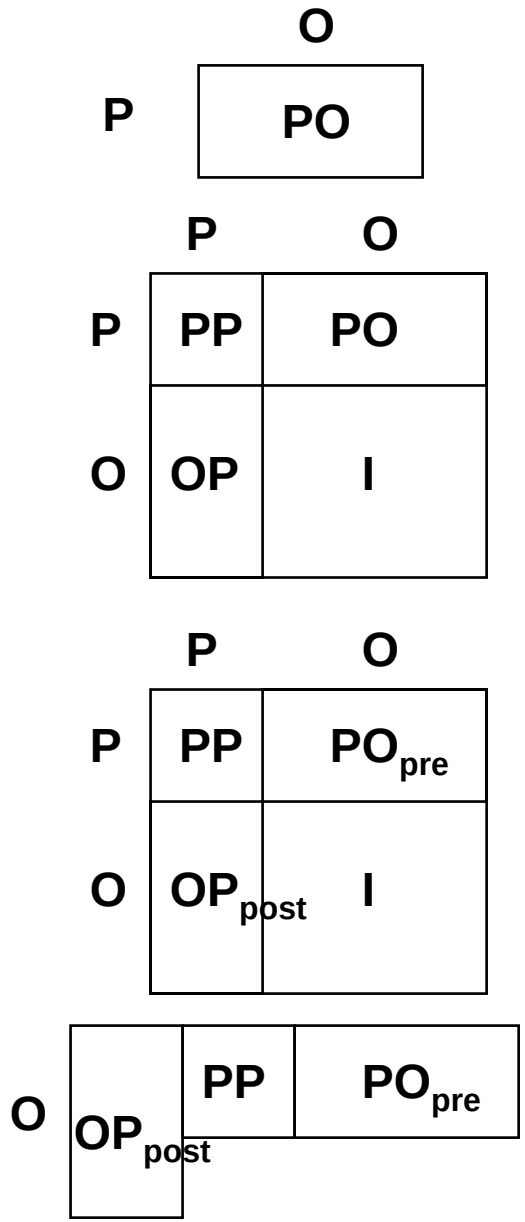
		operator	bump	channel	ramp	step	hole	cavity	star	air	flow	vortex	tone (in)	tone (ex)
Plastic Whistle	creating	a									c			
	transporting			I							e			
	deflecting				I						e			
	venting						I				e			
	creating					I					I	c		
	exciting							I				I	c	
	amplifying						I	I					e	
	coupling						I			I			I	c

		lungs	throat	tongue	lips	mouth	air	flow	vortex	tone (in)	tone (ex)
Human Whistle	creating	a						c			
	transporting		a			a		e			
	deflecting			a				e			
	venting				a			e			
	creating				a			I	c		
	exciting					a			I	c	
	amplifying				a	a				e	
	coupling				a		I			I	c

Non-symmetric, causal, with full NNV structure
A = affects, I = instrument, C = creates, E = effects

Creating Object - Suppressed Process Models

- **Create the Process-Object matrix and insert in true N-square with processes and objects on both side**
- **Move the post process objects (effect and yield) to the lower off-diagonal block transpose**
  - **Leave the pre process objects (consume, agent, instrument) in the upper off-diagonal block**
- **Create the product,and convert back to graph**



Step 1 - Process Object Model

- **The process - object model is created and inserted in a true N square diagram with processes and objects on both sides**

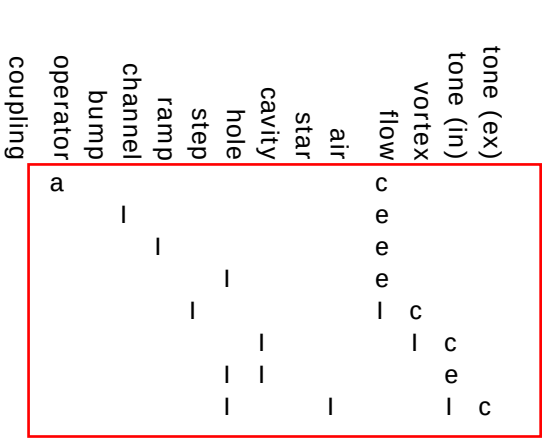
Plastic Whistle

creating
transporting
deflecting
venting
creating
exciting
amplifying
coupling

operator
bump
channel
ramp
step
hole
cavity
star
air

flow
vortex
tone (in)
tone (ex)

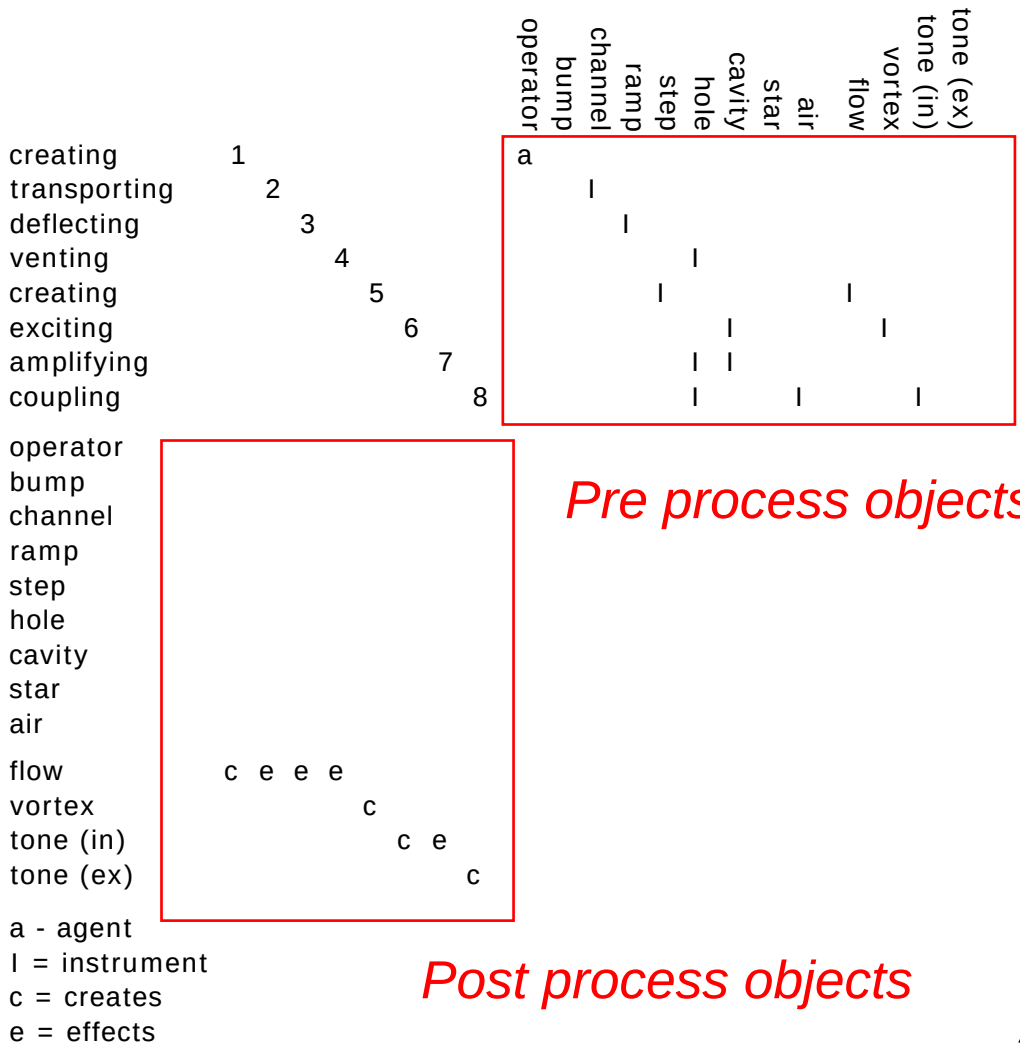
a - agent
I = instrument
c = creates
e = effects



What we had

Step 2 - Move OP Elements

- **Put the post process objects in the lower block (creates, effects)**
- **Leave the pre-process objects in the upper block (consumes, instrument and agent)**
- **Number processes for convenience**



Step 3 - Rearrange Matrix and Form Product

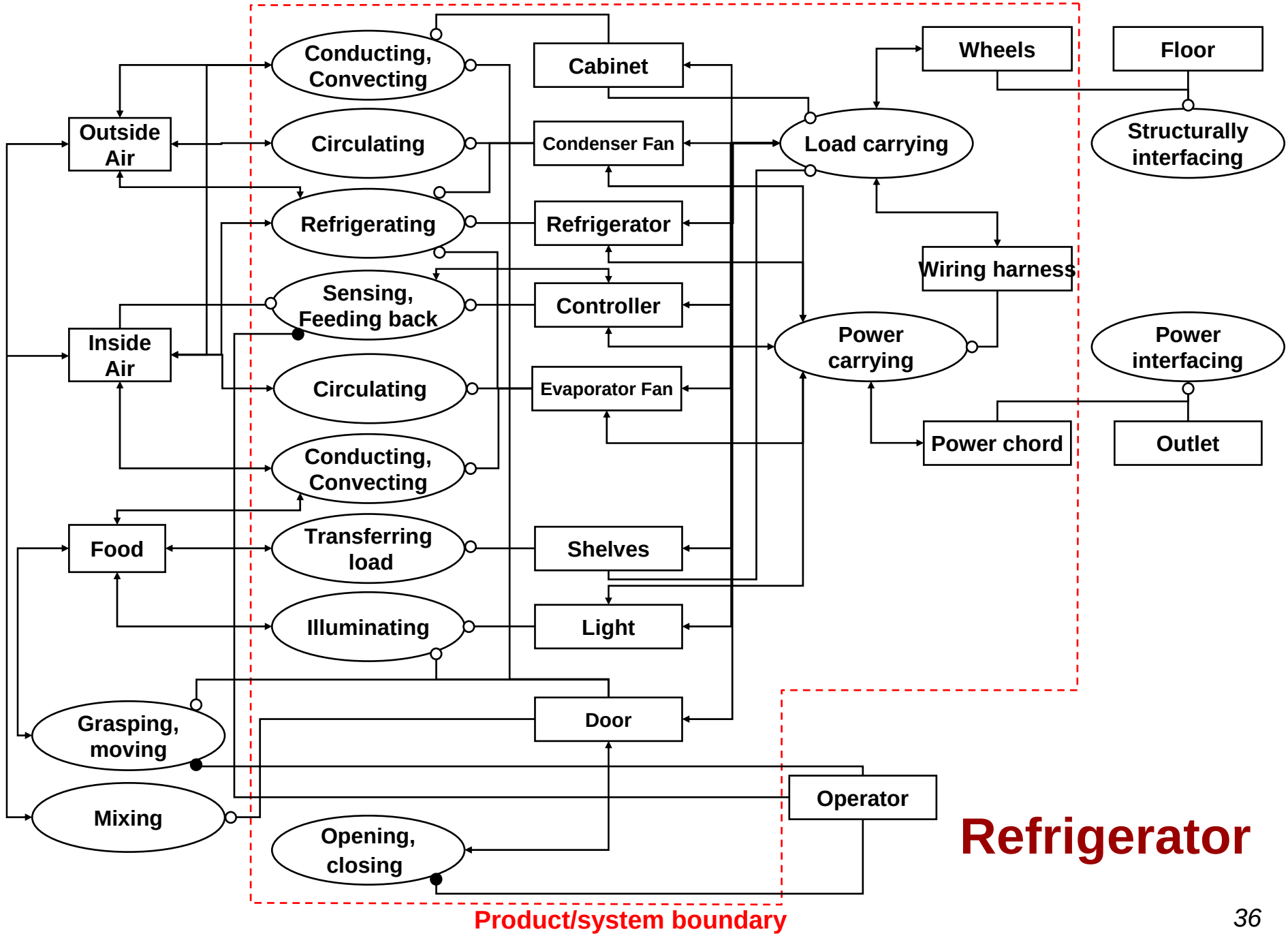
operator	0	0	0	0	0	0	0	0	0	0	0	0	0
bump	0	0	0	0	0	0	0	0	0	0	0	0	0
channel	0	0	0	0	0	0	0	0	0	0	0	0	0
ramp	0	0	0	0	0	0	0	0	0	0	0	0	0
step	0	0	0	0	0	0	0	0	0	0	0	0	0
hole	0	0	0	0	0	0	0	0	0	0	0	0	0
cavity	0	0	0	0	0	0	0	0	0	0	0	0	0
star	0	0	0	0	0	0	0	0	0	0	0	0	0
air	0	0	0	0	0	0	0	0	0	0	0	0	0
flow	c1a	0	e2i	e3i	0	e4i	0	0	0	0	0	0	0
vortex	0	0	0	0	c5i	0	0	0	0	c5i	0	0	0
tone (in)	0	0	0	0	0	e6i	c6i,e7i	0	0	0	c6i	0	0
tone (ex)	0	0	0	0	0	e8i	0	0	e8i	0	0	c8i	0

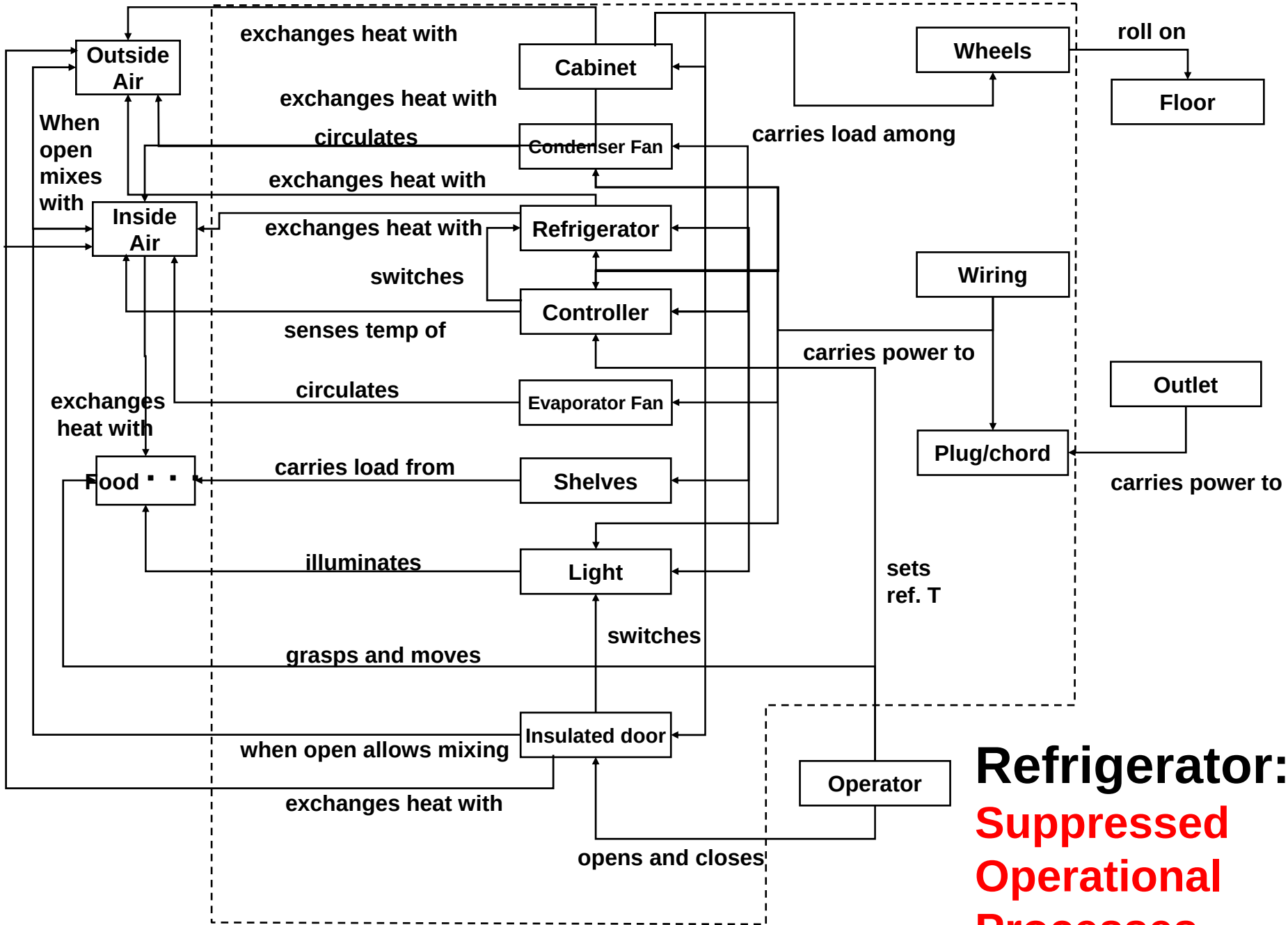
a - agent
I = instrument
c = creates
e = effects

- **Form product OP_{post} *PP*PO_{pre}**
- **This yields a causal, non-symmetric N-squared representation, which you will come to know as a Design System Matrix or DSM**

Matrix Object/Suppressed Operational Processes

	Operator	Bump	Channel	Ramp	Step	Hole	Cavity	Star	Air	Flow	Vortex	Tone in	Tone out
Operator													
Bump													
Channel													
Ramp													
Step													
Hole													
Cavity													
Star													
Air													
Flow	Creates		Transport	Deflects		Vents							
Vortex					Creates					Creates			
Tone in						Excite	Excite/ Amplify				Excites		
Tone out						Couple			Couple			Couple	

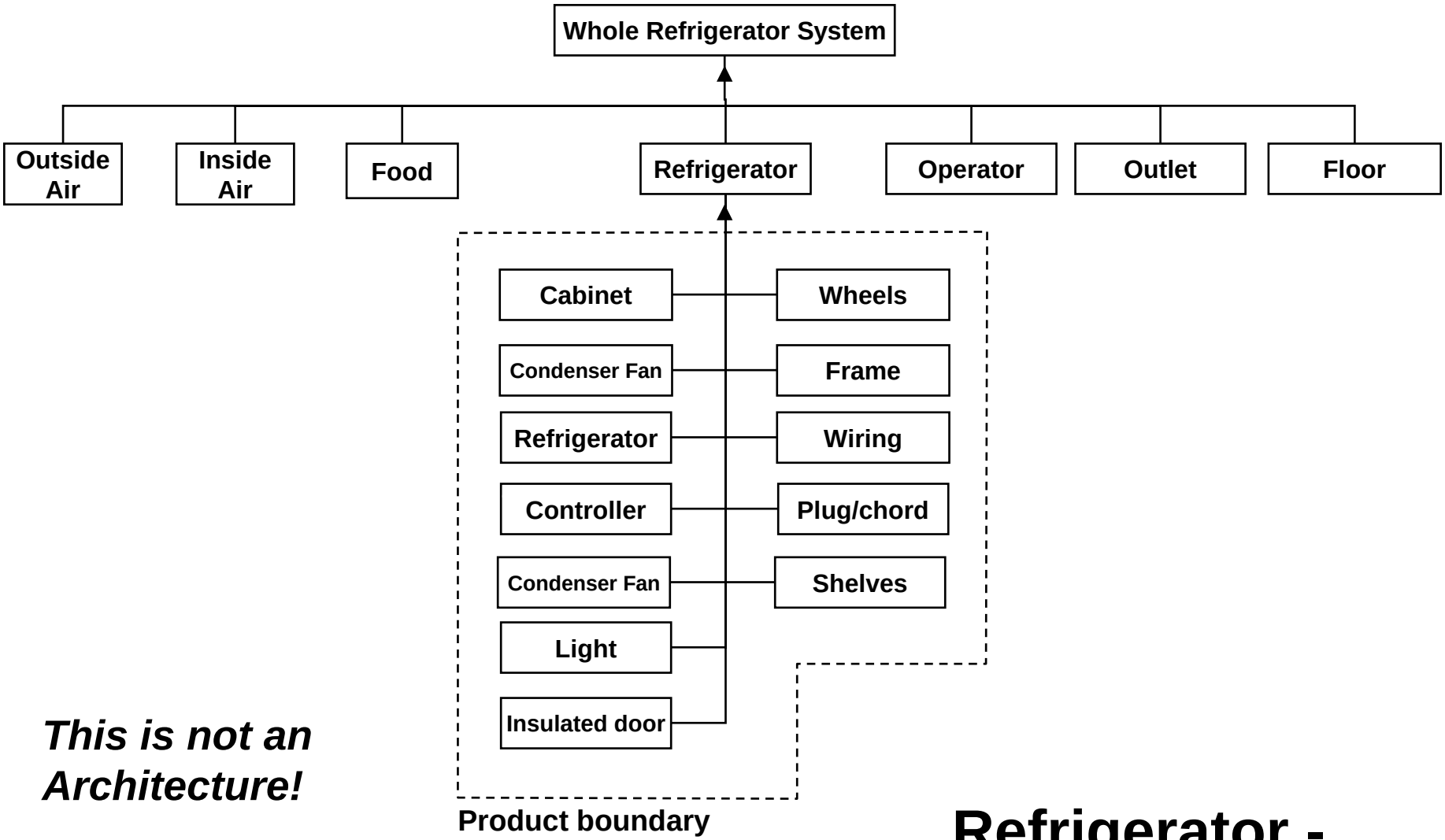




Refrigerator: Suppressed Operational Processes

Alternative Model - Objects Only

- **Finally, the most simplified model is one which removes all information on process and simply shows the objects in a decompositional hierarchy**
- **This is the first model people try to build**
- **Having seen the others, it is obvious just how little information is in this model**



This is not an Architecture!

Refrigerator - Decompositional

Summary - Representations of Architecture

- **Architecture can be represented by matrix/list or graphical representations - in principle the two are interchangeable and contain the same information**
- **Objects and processes can be shown explicitly**
- **One of the other can be suppressed, carefully, for clarity and simplicity**
- **This is much easier for “goes in to, goes out of” architectures than for the more general case**

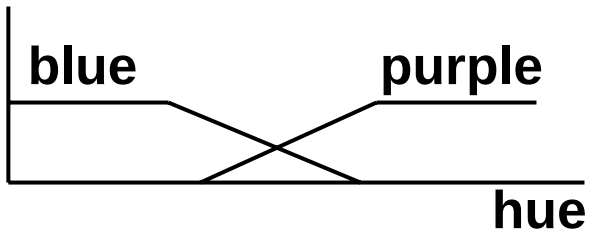
Ambiguity

- **Ambiguity defined**
- **Resolving ambiguity**

Ambiguity

Fuzziness:

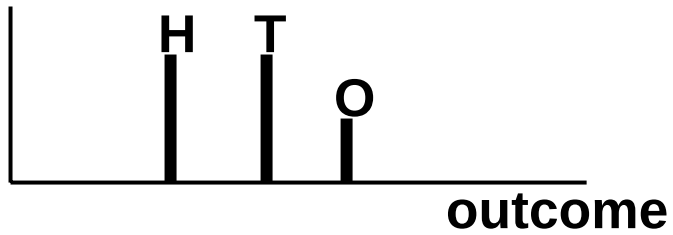
an event or state is subject to multiple interpretations



Fuzzy logic
Fuzzy set theory

Uncertainty:

an event is doubtful or uncertain as to its outcome



Combinatorial analysis
Statistics

Compounding Ambiguity

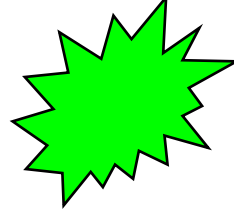
- **Unknown information:**
  - **don't have all of the information**
  - **Information is underdetermined**
  - **You may know, or not know of the existence of the unknown information (unknown unknowns)**
- **Conflicting information:**
  - **have two or more conflicting pieces of information**
  - **Information is over-determined**
- **False information:**
  - **You think you have all the information**
  - **Some of it is wrong**
  - **Information appears determined**

- X , __ , Z
- X , Z
- X_p,X_i , Y , Z
- P , Y , Z

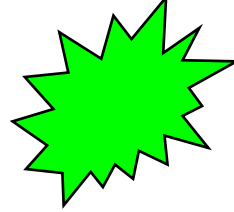
Examples of Ambiguity

- **Please make me a smooth cover for this phone**
- **Will it be a boy or a girl?**
- **Make sure you meet your quarterly goals**
- **Produce a low cost, high quality product**
- **Every fourth year is a leap year**

Ambiguity in Design Challenge 2?



Ambiguity in Coin Flip



- How does an NFL game begin?
- Flip the coin and report heads or tails
- Both type of ambiguity are present here!!!!

Approaches to Resolving Ambiguity

- **Definitions and language in common [SA]**
- **Frameworks for organizing thinking [products - SA, processes - SE, organization - SPM, OP]**
- **Standardized processes [SE, SPM]**
- **Exercises to “sharpen” upstream influences [SA, Marketing]**
- **Defining system boundaries [SA]**
- **Planning for robustness [SE, Robust Eng.]**

Summary - Ambiguity

- **Ambiguity stems from multiple interpretations and uncertainty, compounded by incomplete, conflicting or incorrect information**
- **The beginning of the PDP - the interface with the upstream process - is a time of great ambiguity**
- **The role of the architect is to resolve this ambiguity so as to create the environment in which the PDP can quickly and successfully meet its goals.**

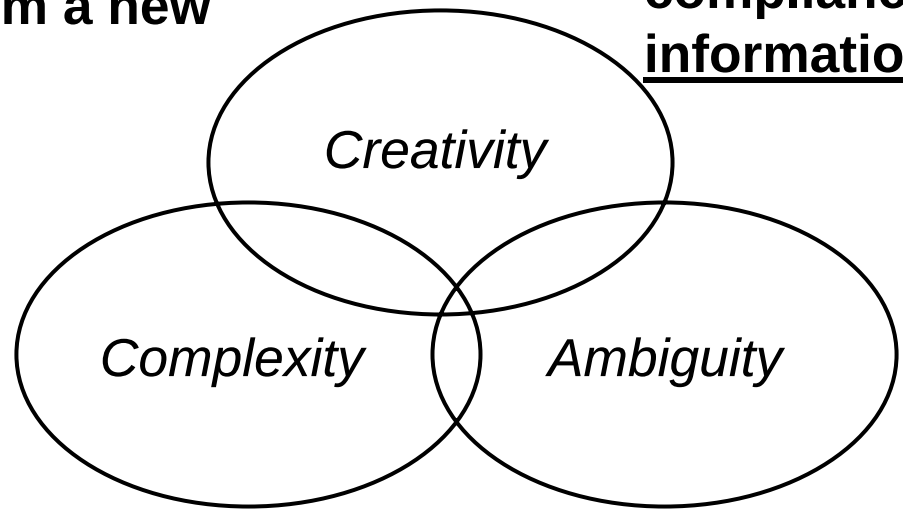
Three Themes in Architecture

- **Ambiguity:**
  - **Susceptible to multiple interpretation**
  - **Doubtful or uncertain**
- **Creativity:**
  - **The ability or power to: cause to exist, bring into being, originate**
- **Complexity:**
  - **Consisting of interconnected or interwoven parts**

Source: American Heritage Dictionary

Creativity - The innovative combination of different pieces of information or creation of new information required to to form a new system

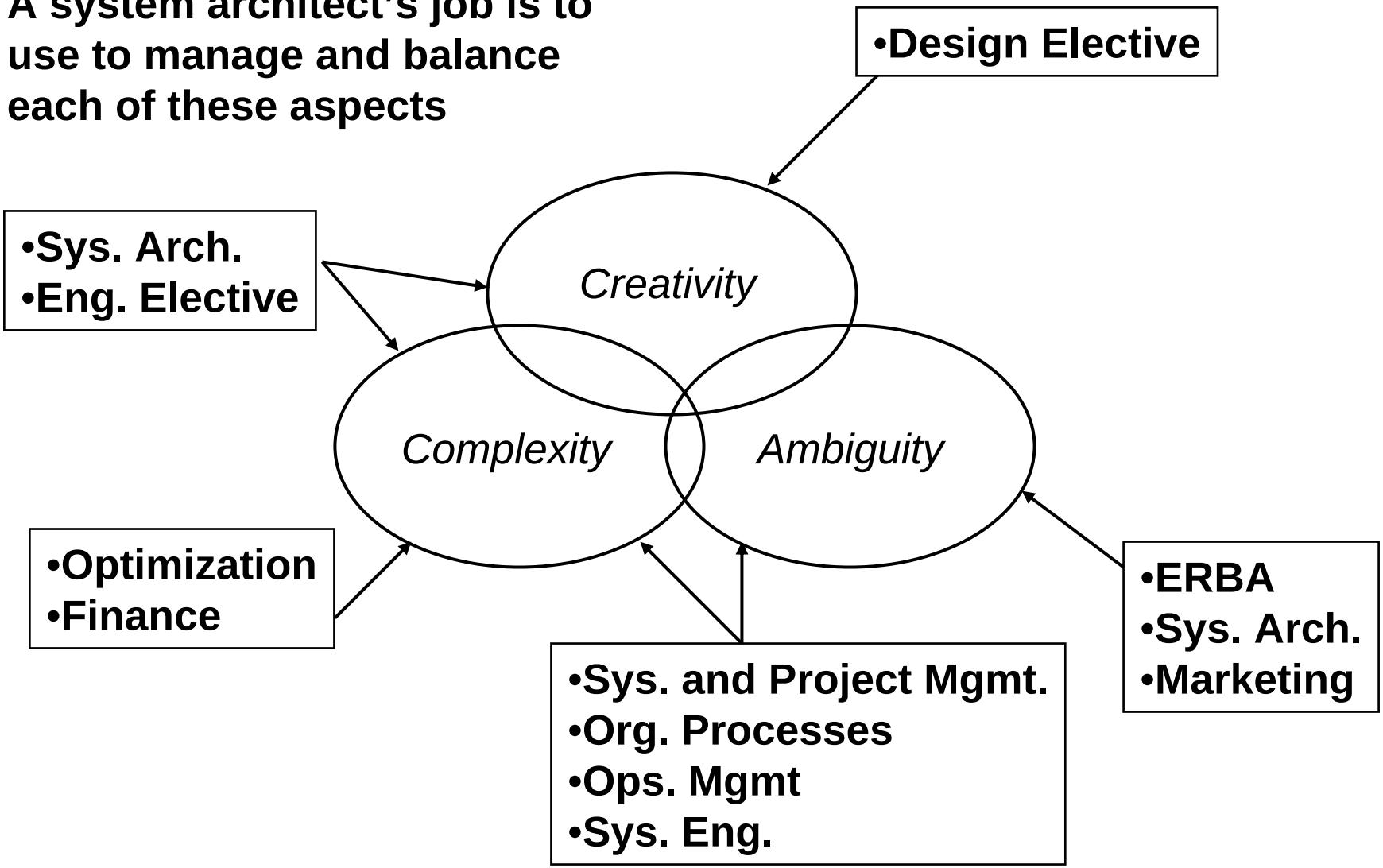
Ambiguity - The uncertainty associated with numerous choices, unclear objectives or indefinite functional compliance = imprecise information



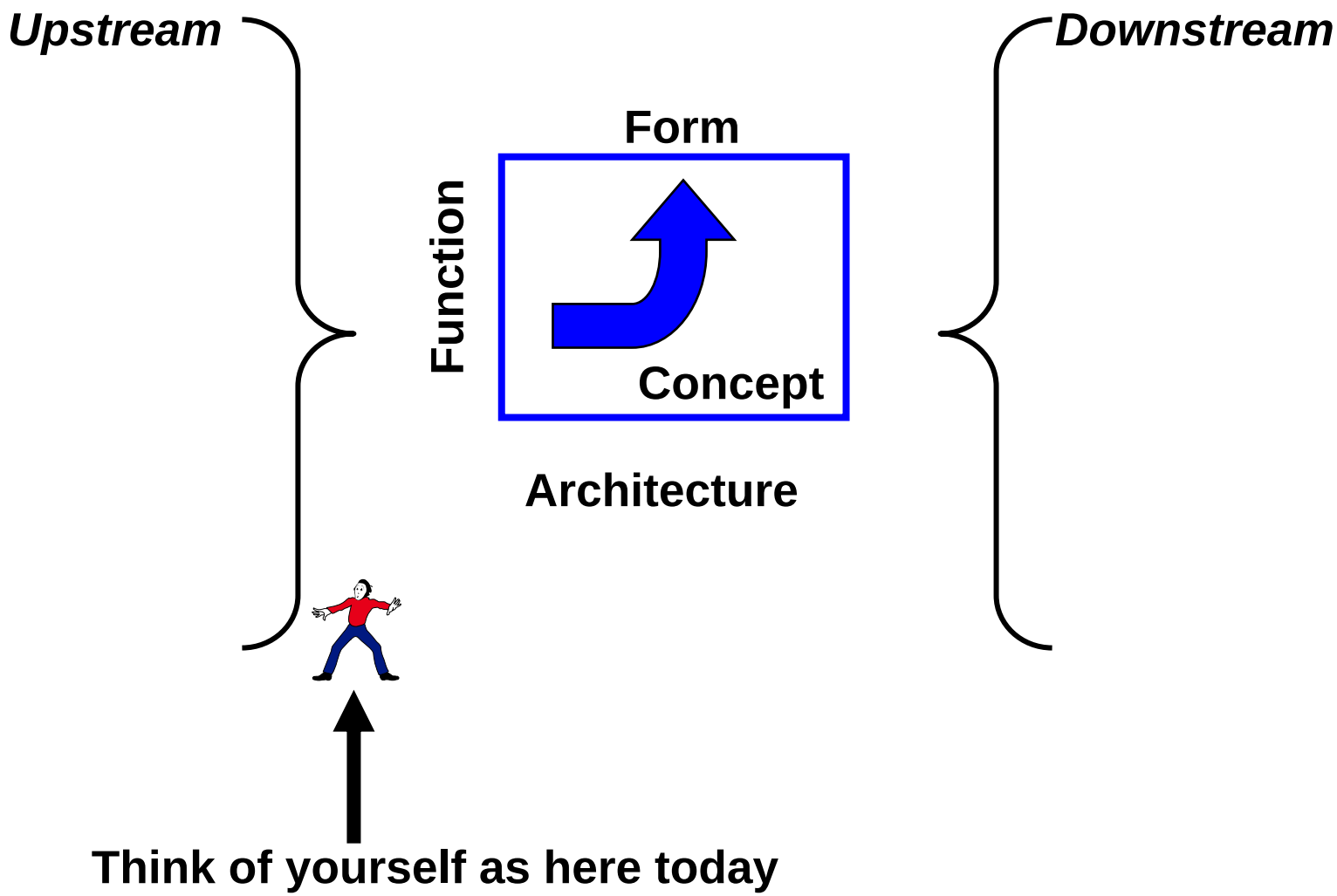
Complexity - The amount of information required to fully describe a system and its interfaces

Product Development - The transformation of a vector of ambiguous and imprecise information (upstream) into a vector of precise information describing the product (the information object known as "the design")

A system architect's job is to use to manage and balance each of these aspects



Architecture Centric View of the PDP



Upstream Ambiguity in Architecting

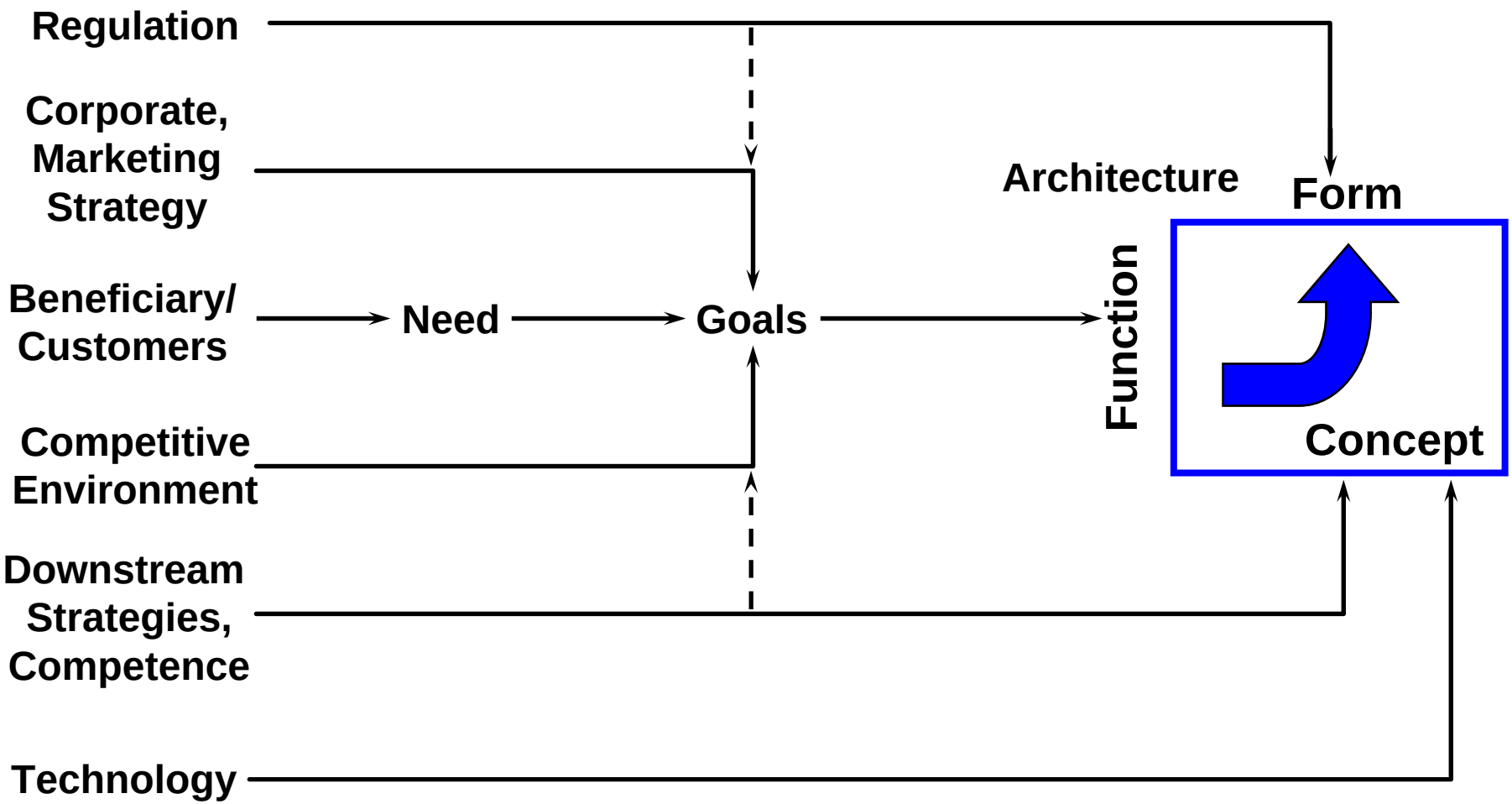
- **What is our objective?**
- **How far do our scope and responsibility range?**
- **How much risk is (corporate, Washington, the board) willing to take?**
- **Will marketing buy it?**
- **What does the customer want/need?**
- **Will our requirements change with time?**
- **Is that technology infusible?**
- **What are applicable regulations? Are they likely to change?**

Removing/Reducing/Resolving ambiguity is a main role of the architect at the interface with the upstream process

Ambiguity in the Upstream Influences

- **No one explicitly "designs" the upstream influences, rather they just occur, often with incomplete, overlapping, or conflicting outcomes**
- **The architect must engage these upstream influences and drive the ambiguity out to create a vision and plan for a successful product**
- **This requires knowledge of**
  - **What the upstream influences are, who has "control" of them, and how they are engaged**
  - **What the object is - a consistent, complete, attainable, clear and concise set of goals for the product**
  - **How to get from A to B**

Dominant Upstream Influence on Architecture

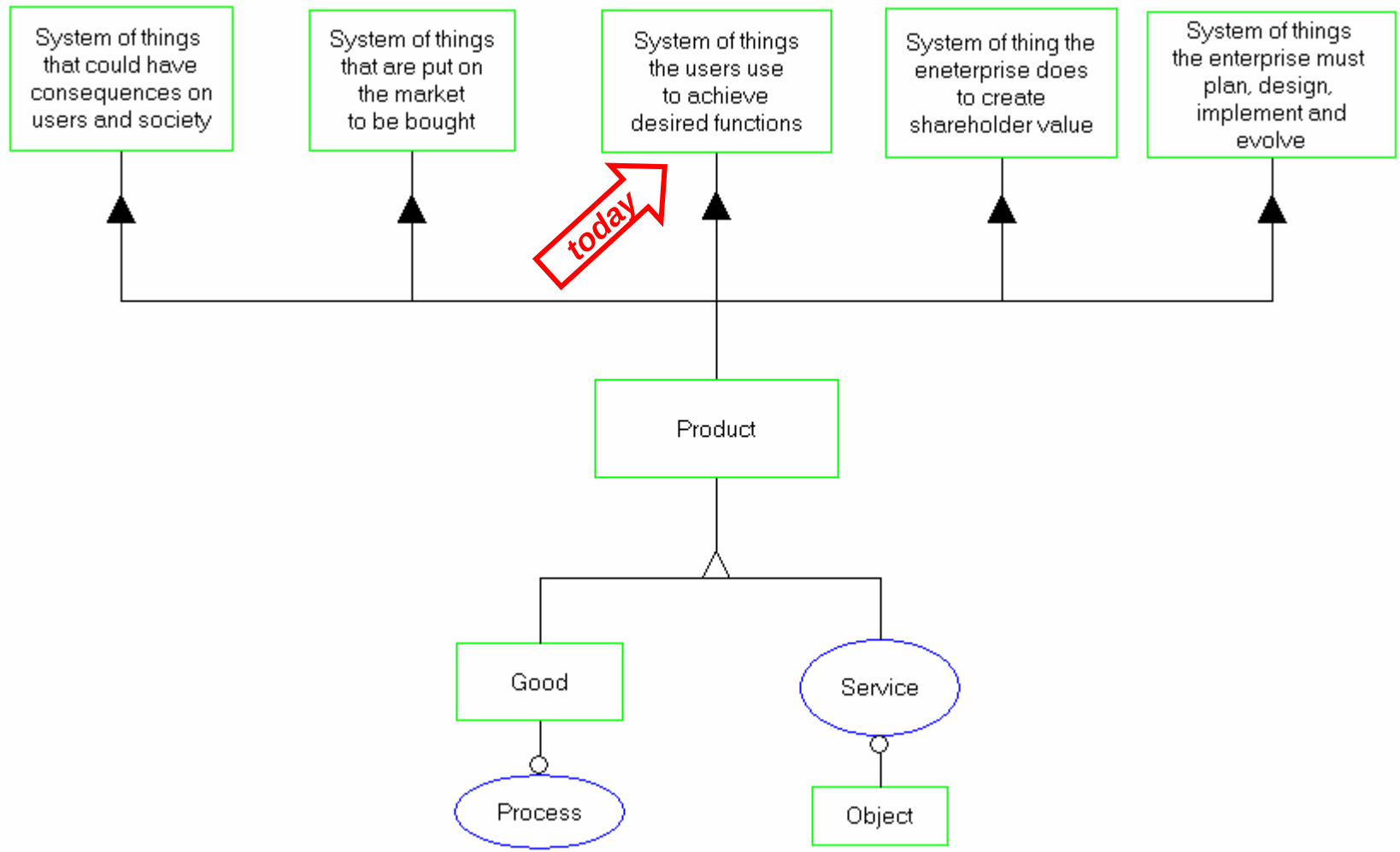


Principle Upstream Influences

- **The customers and their needs**
- **The corporation, which has corporate strategy and functional strategies (marketing, etc.) which define what the company does, how it competes, what its values are, what the return to investors will be, etc.**
- **The market, with its forces and competitors**
- **Regulations and similar pseudo-regulatory influences such as case law, impending regulations, and standards**
- **Technology, which is or will be available and infusable in your system**
- **The competence, team, facilities, processes which will be employed downstream (planning, designing, implementing, operating and evolving), but must be considered at a strategic level as an upstream influence**

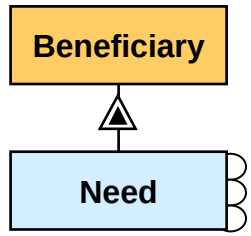
These things all fundamentally influence architecture

Product/Systems and Their Super-Systems



Need - Defined

- Need is a product attribute
- Need is defined as:
  - a necessity
  - an overall desire or want
  - a wish for something which is lacking
- Can also include opportunities to fill unexpressed or unrecognized needs



▲ Decomposes to
△ Specializes to
◬ Has attribute of

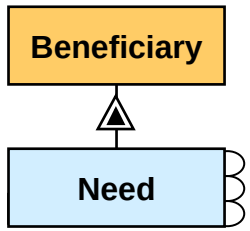
Needs - Described

- **Exist in the mind of the beneficiary**
- **Primarily *outside* of the producing enterprise**
- **Expressed often in fuzzy or general (i.e. ambiguous) terms**
- **Are interpreted (in part) by the architect**
- **Who has relevant needs?**

- **Start by focusing on the needs of the direct beneficiary!**

Value Questions (1)

- **What is the mission of our enterprise, and the outcomes**



- **Who is the beneficiary?**
- **What is their need(s)?**

Need

▲ Decomposes to
△ Specializes to
◬ Has attribute of

Value Identification - Refrigerator

- **We are a for-profit enterprise in the home appliance business**

- **Who is the beneficiary?**
- **What is the need?**

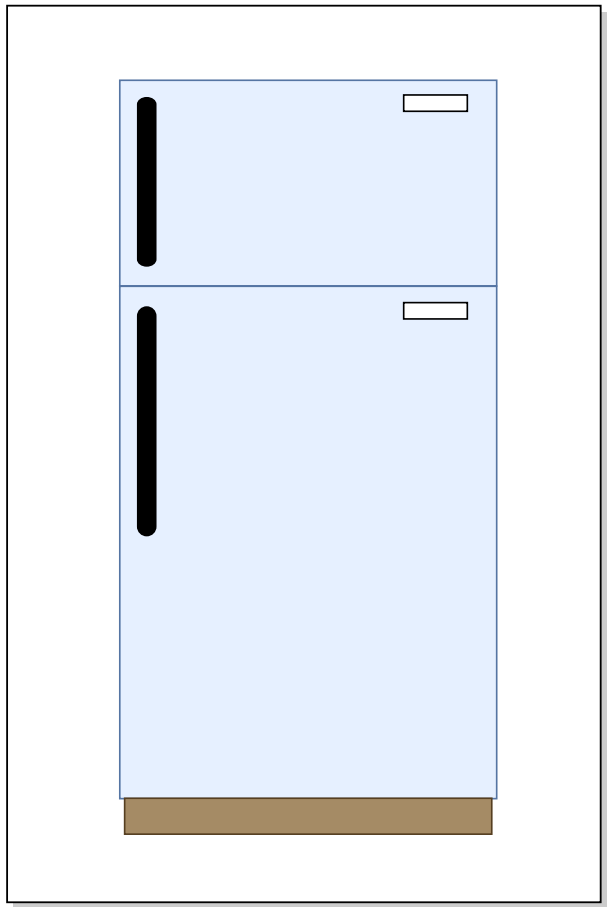
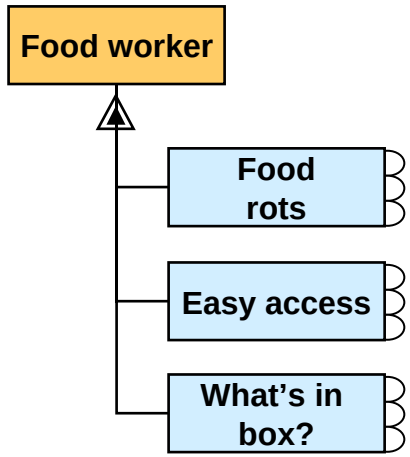


Figure by MIT OCW.

Multiple Needs - Refrigerator

- **Focus first on the primary beneficial stakeholder - the food worker**
- **The food worker has multiple needs**
- **Enumerate needs:**
  - **Spoilage**
  - **Access**
  - **Data on refrigerator contents**



▲ Decomposes to
△ Specializes to
◭ Has attribute of

Needs - Who has Them?

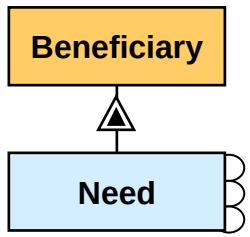
- The important needs are those of the ***beneficiary***
- In simple and consumer systems, the beneficiary is also often the customer - the person or entity who is on the receiving end of the transfer associated with the product
  - Notable exceptions: presents, buying for the household/family
- In more complex products, the customer is often not the beneficiary
- If the beneficiary is not the customer, the customer plays a role in interpreting the needs of the beneficiary
- User or operator (the person who operates the product) also has needs that will appear through operator
- In simple and consumer system, the user is also often the customer

Use Context

- **A good way to understand needs is to examine the use context - that set of objects and processes that make up the current (or planned) environment in which the product/system will operate**
- **Examining usage context will often help identify value**
- **Examining usage will sometimes reveal latent needs - needs which the beneficiary may not realize they have or be able to verbalize**

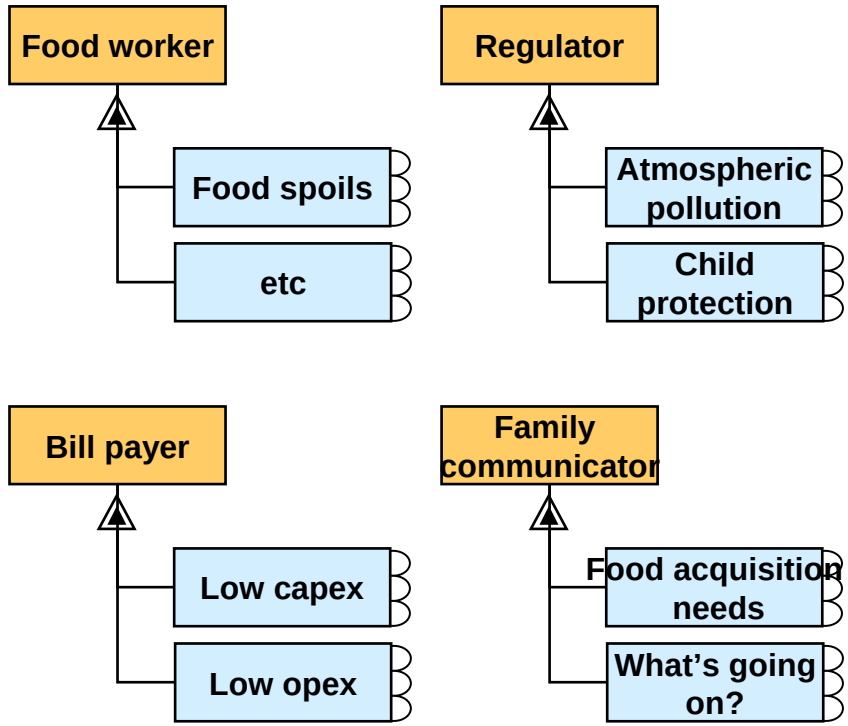
Representing Beneficiary Needs

- **Beneficiary is an object who is a person (shaded generic skin color to remind us its of human nature)**
- **Need is a characteristic of the beneficiary (with a light blue shading, and clouds to the right, to remind us of the vaporous and fleeting nature of a need)**
- **To identify needs, identify beneficiaries, and then their needs (from use case)**



Multiple Beneficiaries - Refrigerator

- **Food worker is not the only one who benefits from a refrigerator**
- **Identify the important beneficiaries**
  - **Who benefits?**



Decomposes to
Specializes to
Has attribute of

Summary - Needs

- **Needs exist in the heart and mind of the beneficiary**
- **They exist outside the enterprise**
- **They are fuzzy, ambiguous and ill stated**
- **They must be identified and understood**

- **A beneficiary often has more than one need to be met by a product system**
- **There is often more than one beneficiary**

Goals - Defined

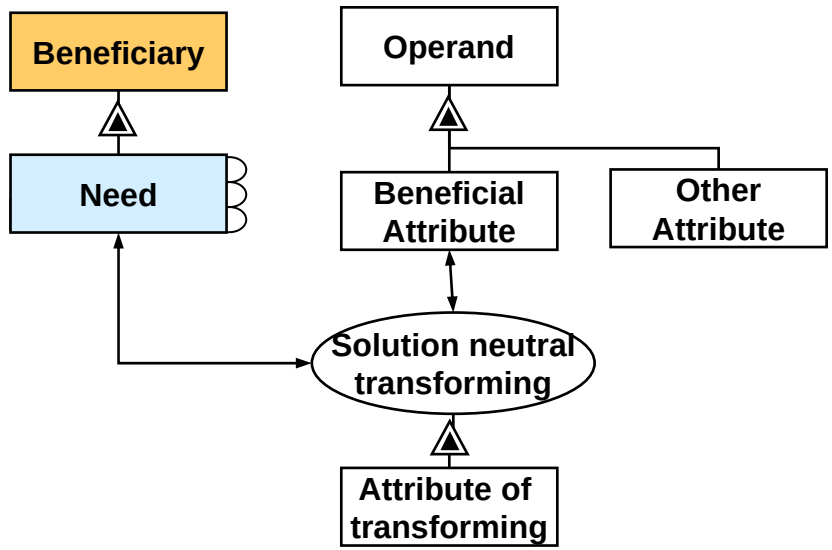
- **Goal is a product attribute**
- **Goal is defined as**
  - **what it *planned* to be *accomplished***
  - **what the *designer* hopes to *achieve* or *obtain***
- **Will include goals derived from beneficiary *Needs* i.e. the primary external functional goals**
- **Will also include goals from corporate strategy, regulations, the competitive marketplace, regulation, etc.**
- **Define how the product will benefit the beneficiary (customer), the enterprise and society**
- **Expressed in (hopefully) precise terms of System Development**

Goals - Described

- **Are defined (in part) by the architect**
- **Goals exist within the enterprise, and are under the control of the enterprise**
- **Goals are often *traded* against Form and Function in design**
  - **Therefore should be considered an independent attribute**
- **Embodied in a statement of goals (specifications?, requirements?, constraints?)**
- **Goals drive metrics to be used in the product/system success criteria**

Value Identification - Goals on Externally Delivered Function

- **Examine the operand associated with value**
- **Identify the attribute of the operand whose change is associated with value**
- **Define the transformation of the attribute associated with value, in solution neutral form and its attributes**



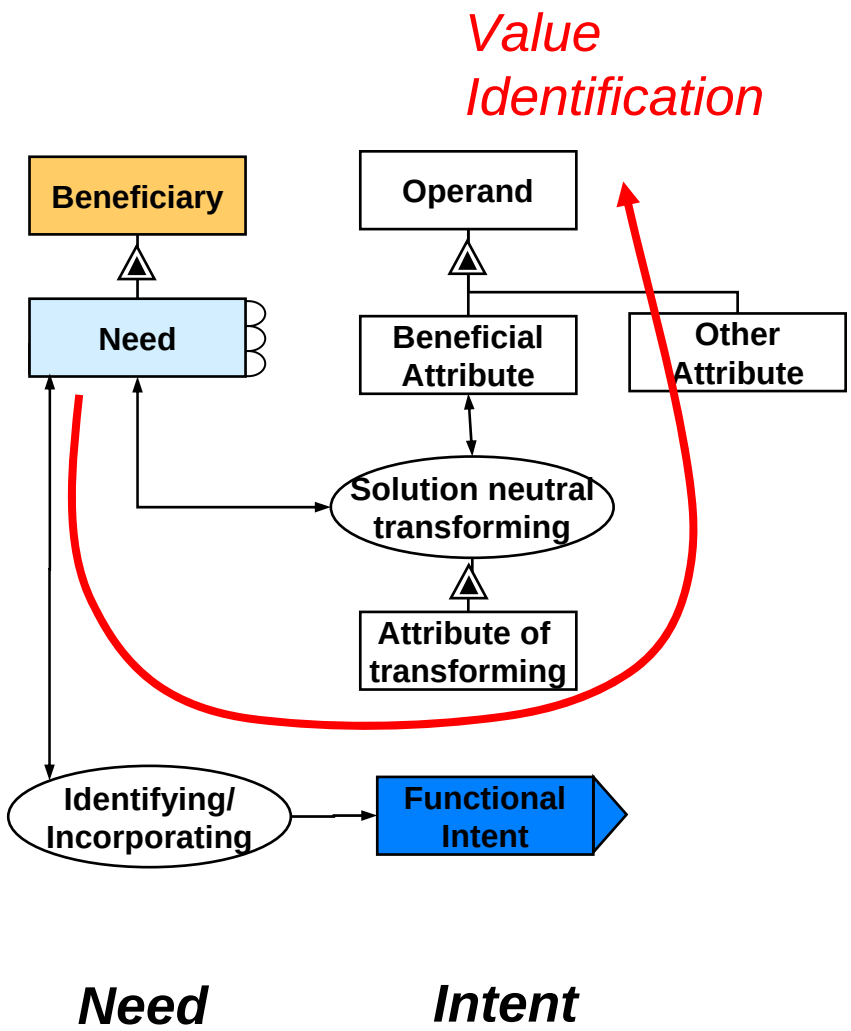
Need ***Intent***

This will lead you to a value focused solution neutral statement of intent on function

▲ Decomposes to
△ Specializes to
▲ Has attribute of

Value Questions (2) - Value Identification

- Who is the beneficiary?
- What is the need?
- What is the value related operand?
- What is the value related attribute?
- What is a solution neutral statement of the value related transformation?
- What are other important attributes of operand and transformation?



Solution neutral statement becomes the intent on function

Value Identification - Refrigerator

- **Who is the beneficiary?**
- **What is the need?**
- **What is the value related operand?**
- **What is the value related attribute?**
- **What is a solution neutral statement of the value related transformation?**

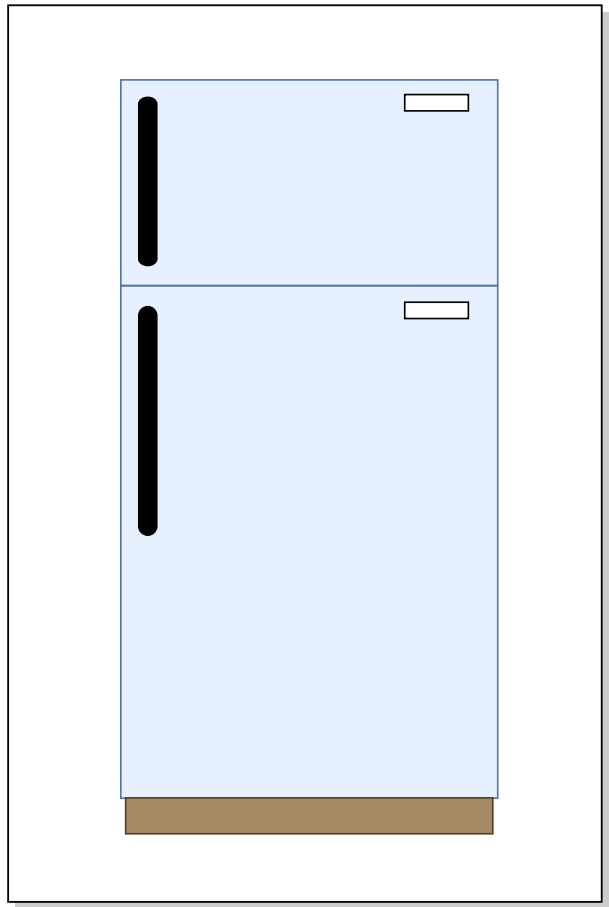
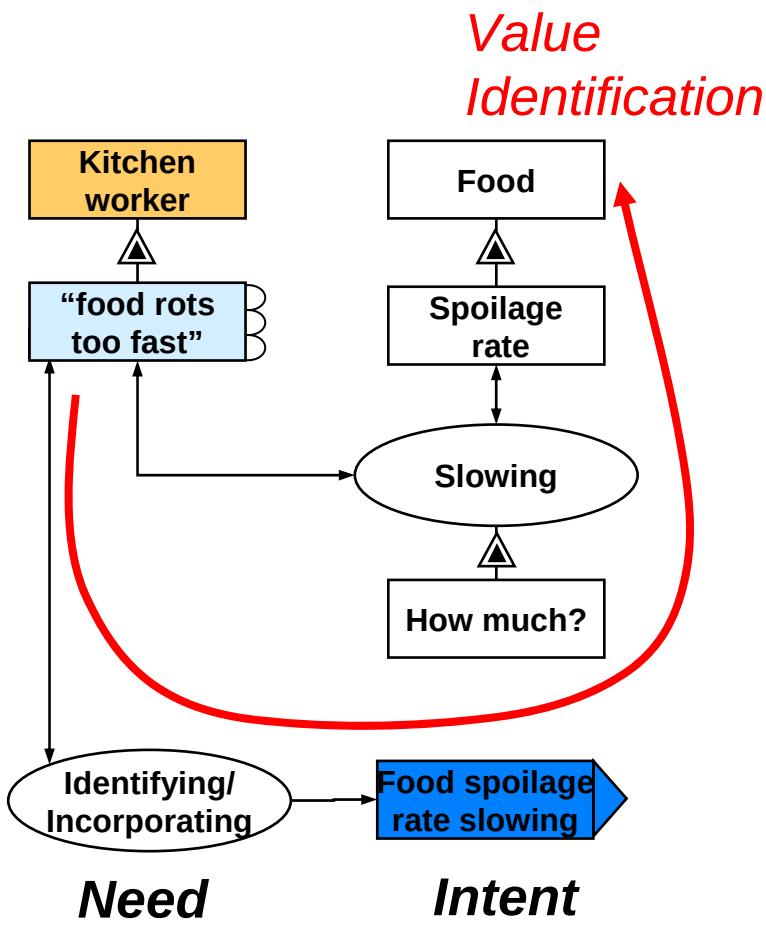


Figure by MIT OCW.

Value Identification - Refrigerator

- Beneficiary = kitchen worker
- Need “my food rots too fast”
- Operand = food
- Value attribute = spoilage rate
- Transformation = slowing

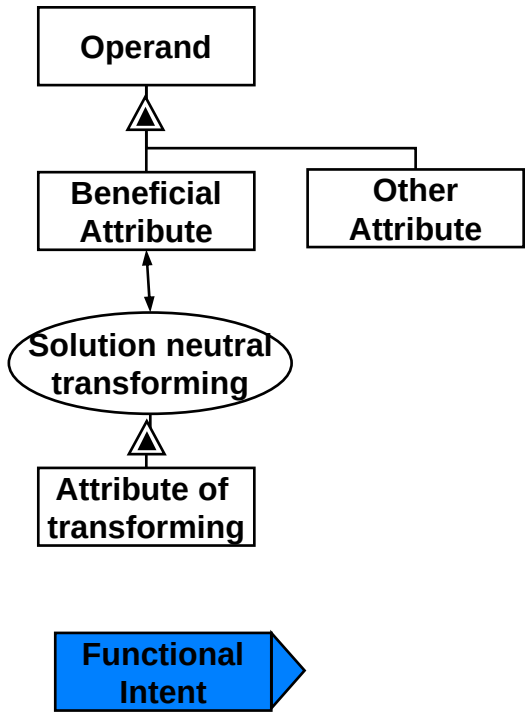


Intent

- An Intent is
  - What the purpose is
  - What someone hopes to achieve or obtain
- Is *always defined by someone*
- The definition of intent is more limited and precise than goal - it is a fragment of the total goal statement
- Useful to create a special symbol for this information object (with a darker blue, indicating firming up of an understanding of need, and with an arrow to remind you of where you are going)

Template for Statements of Functional Goals

- Characterize the Operand
- State the beneficial attribute in the operand that will change
- State other attributes of the operand that are important
- State the transformation
- Stat the attributes of the transformation that are important

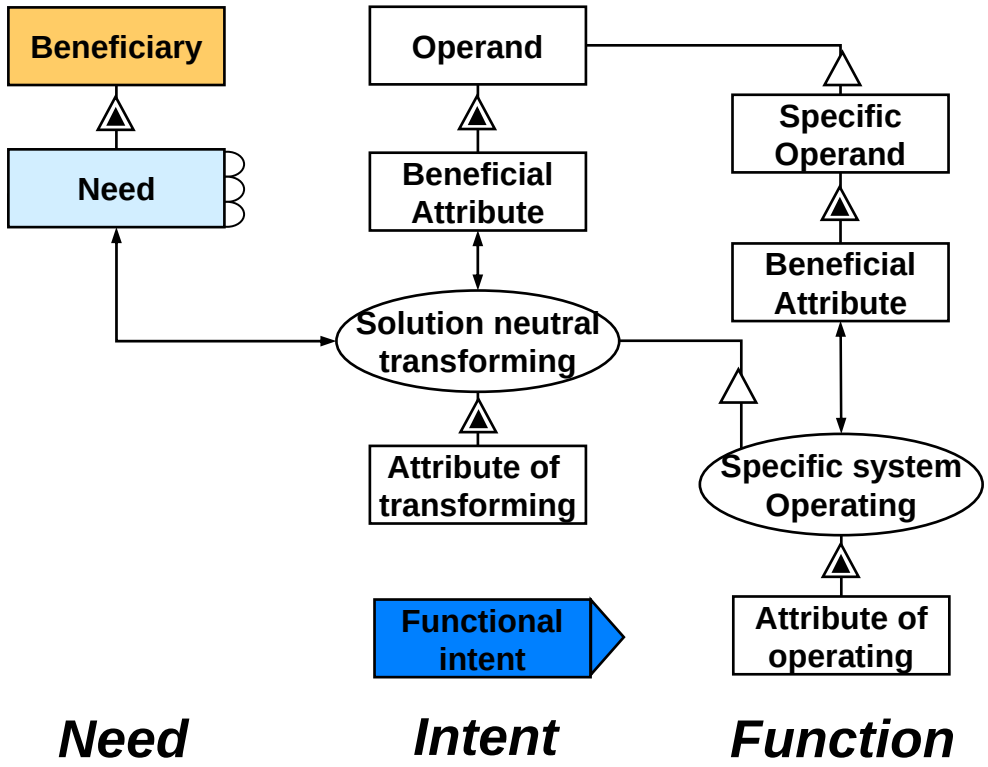


Intent

Solution neutral statement becomes the template for functional goals

Solution Neutral to Specific Function

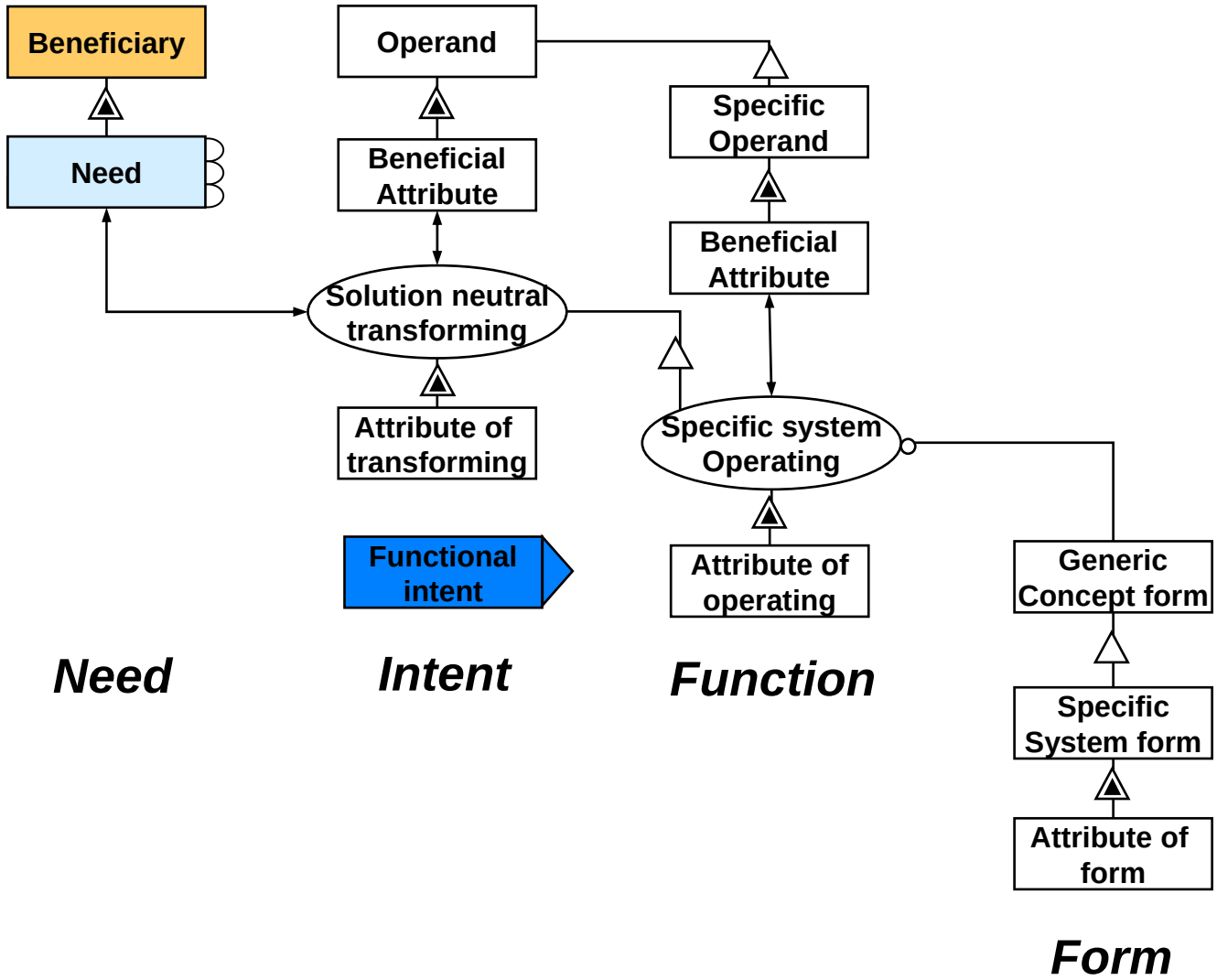
- **Next identify the specific operand (if not the same as the generic operand), and its specific beneficial attribute**
- **Then choose a the process part of the *concept* which specializes the solution neutral process**
- **Define attributes of the process**



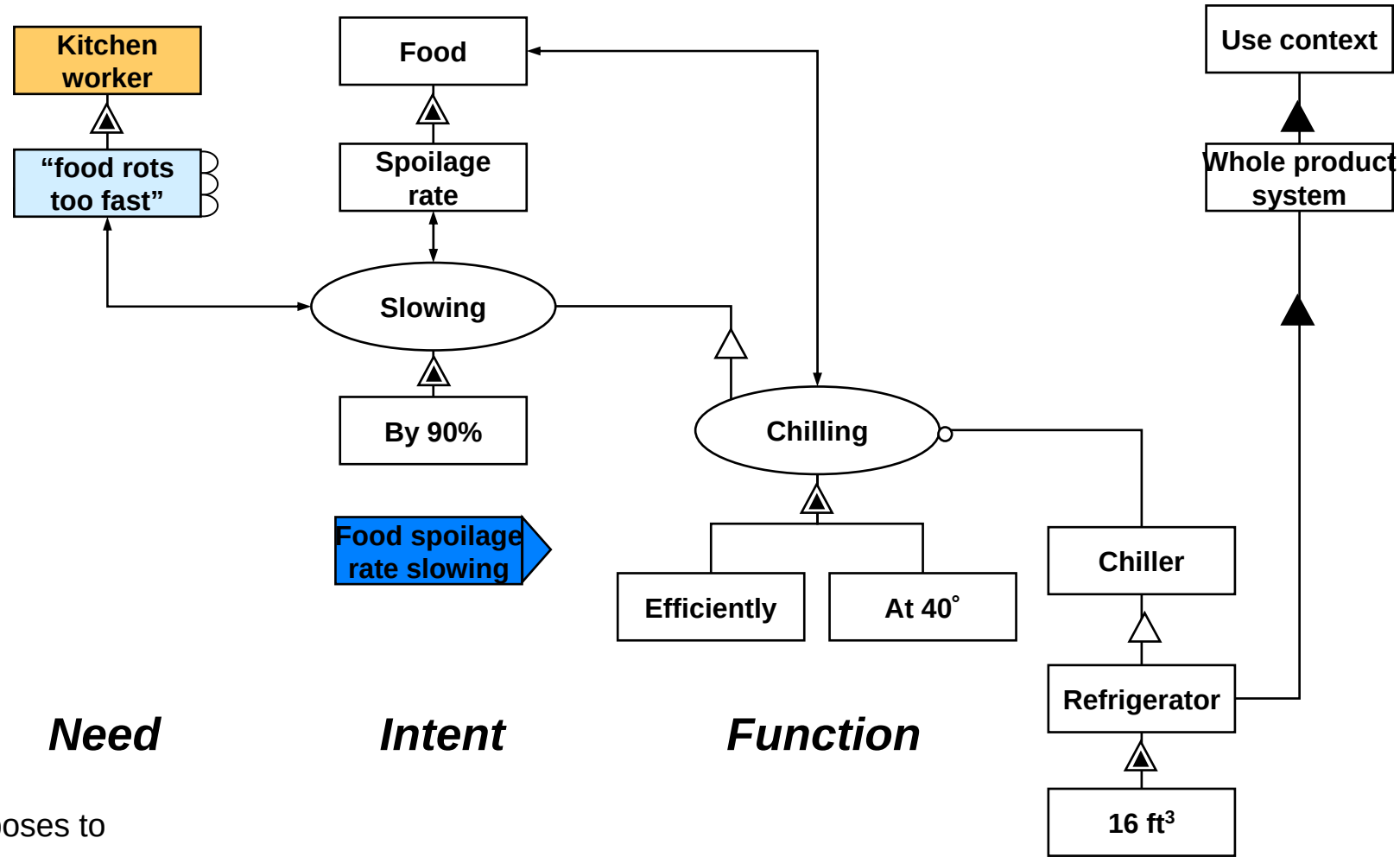
▲ Decomposes to
△ Specializes to
◭ Has attribute of

Specific Function to Specific Form

- Next choose the generic and specific form part of the *concept* to execute the specific process
- Define the attributes of the form



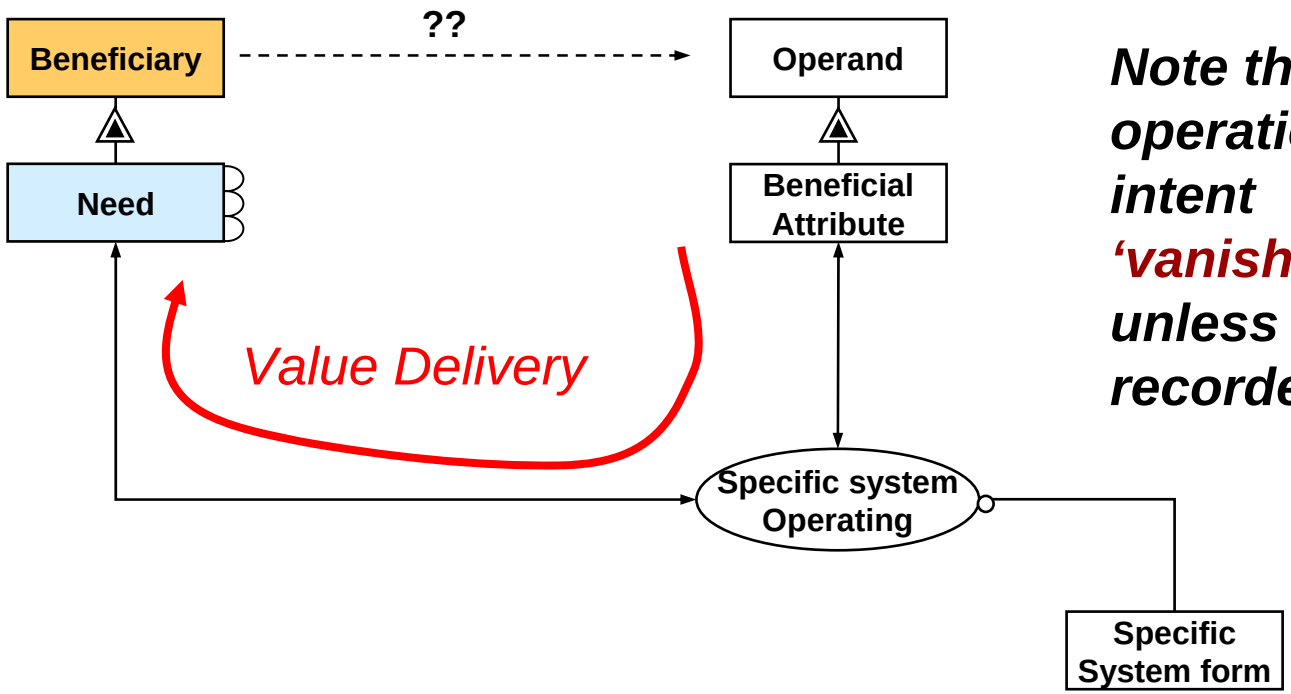
Complete Value Template - Refrigerator



- ▲ Decomposes to
- △ Specializes to
- ◭ Has attribute of

Value - A Formal Definition

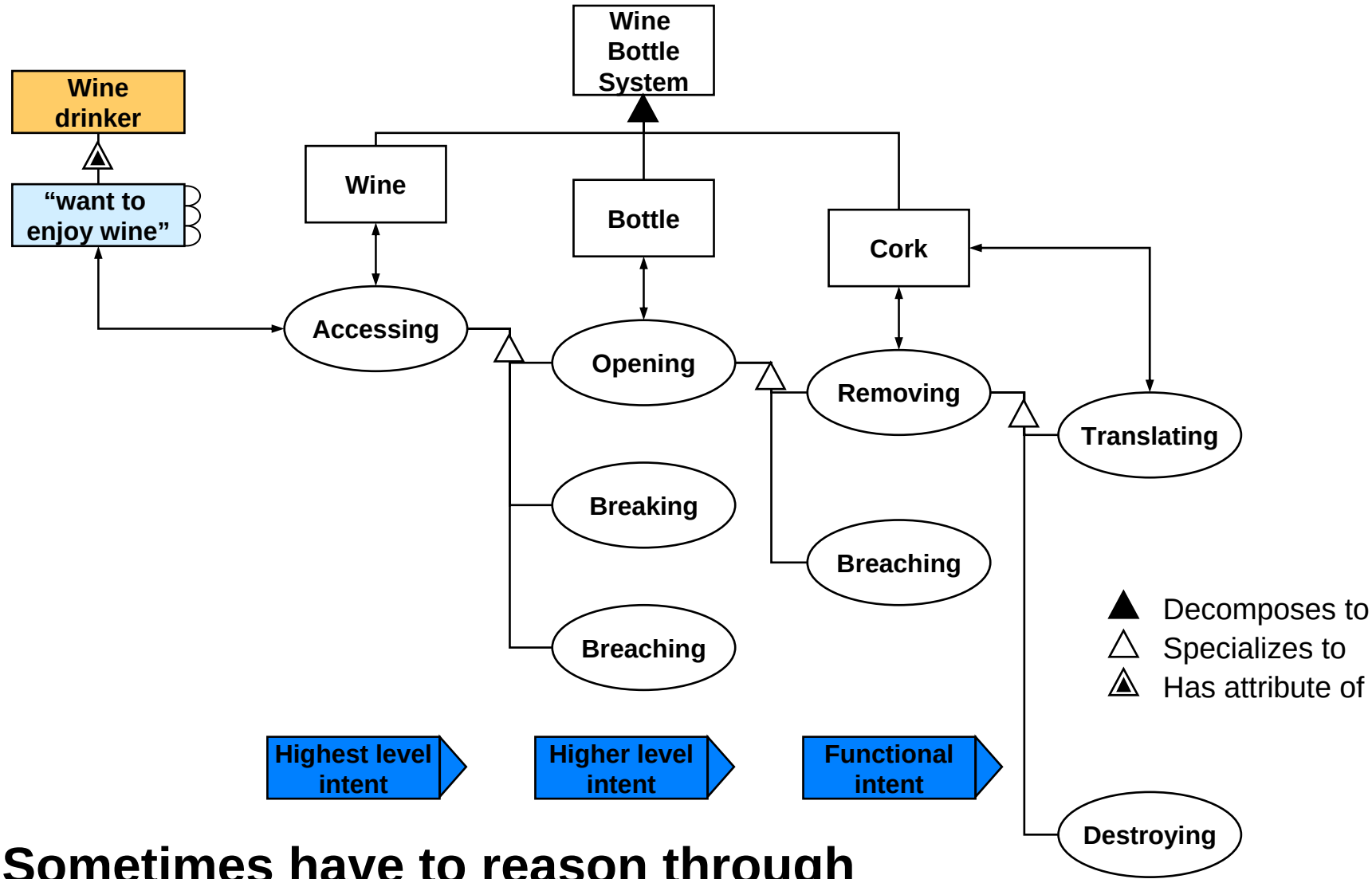
Value is delivered when the external process(es) acts on the operand in such a way that the needs of the beneficiary are satisfied at a desirable cost.



Note that in operations, intent 'vanishes' unless recorded

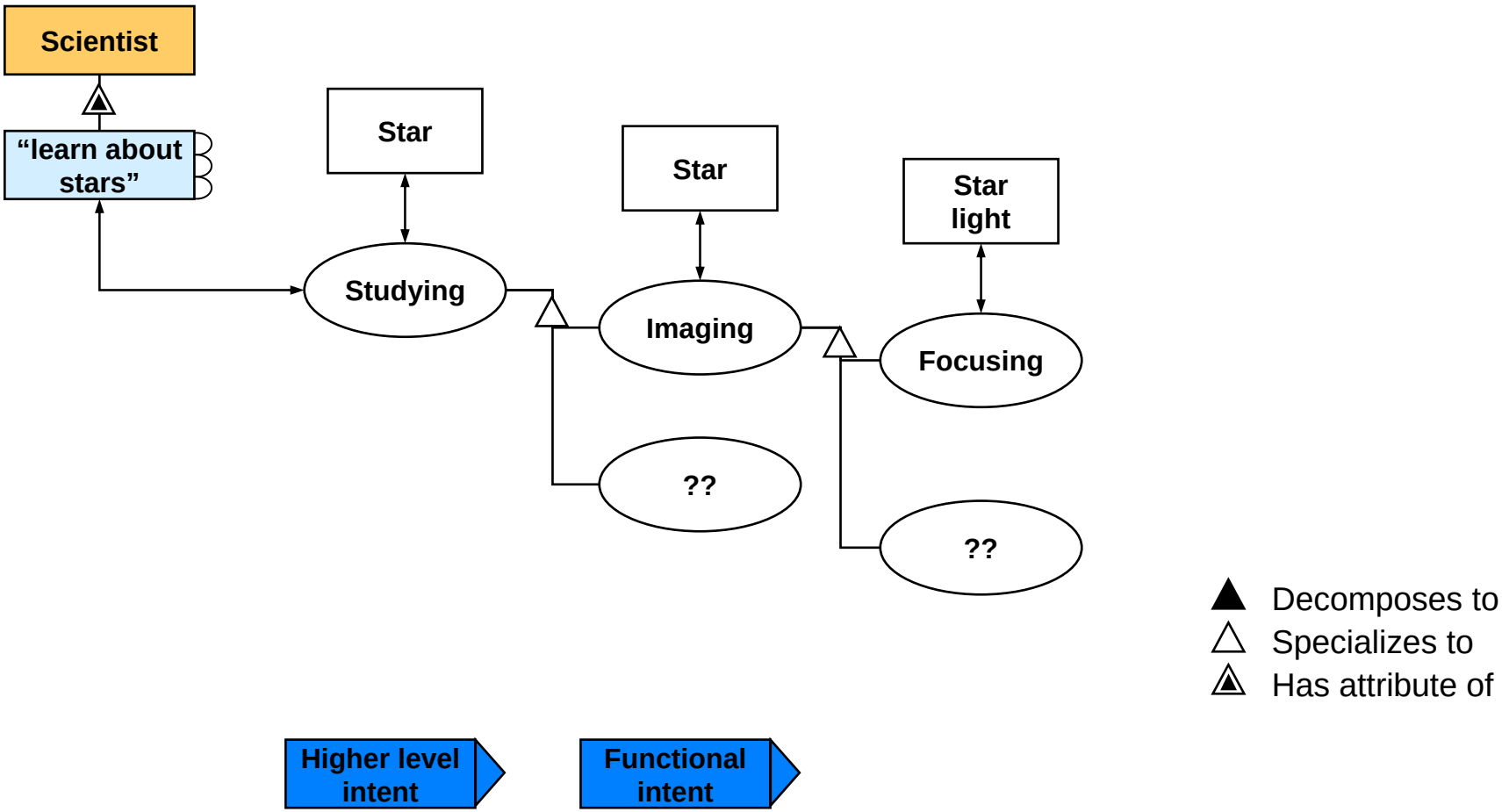
The relation between the beneficiary and the operand in undefined (the beneficiary could be the operand, own the operand, love/hate the operand, etc.)

Levels of Intent - Corkscrew



Sometimes have to reason through several layers of intent

Levels of Intent - Telescope



Sometimes have to reason through several layers of intent

Intent as Operand plus Process?

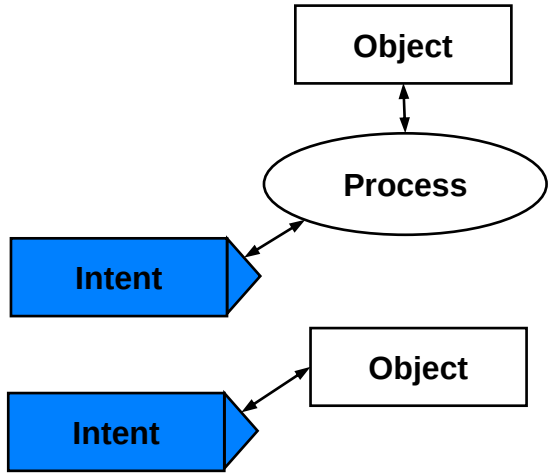
- **So what are the value related operand and processes associated with**
  - **A movie?**
  - **A book?**
  - **Styling of a laptop?**
  - **Driving a BMW SUV rather than a Ford of the same quality and performance (i.e. brand name)?**
  - **Strategic planning?**
  - **Owning a piece of fine art?**
  - **Collecting coins?**
  - **A dress or suit?**

Types of Goals

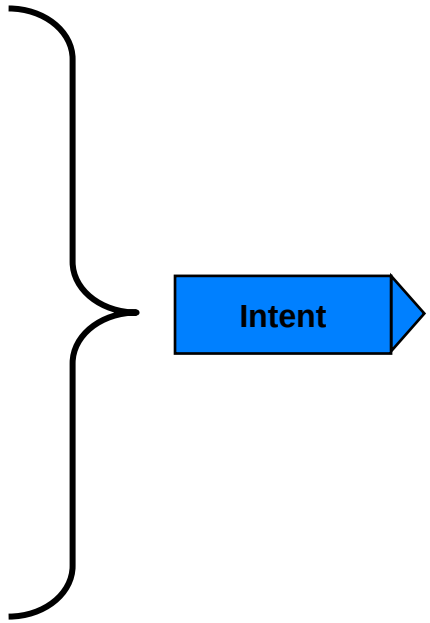
Goals are a mixture of several types:

- **Intent on Function**

- **Intent on instrument Object**



- **Intent on the design process**
- **Intent on the implementation process**
- **Intent on the operation process**
- **Intent on the strategic process**
- **Intent on the market process**
- **Intent on the regulatory process**
- **Intent on the technology process**



See the pattern???

Goals on Function, Form, …

- **Goals can be on externally delivered Function; include adverb or adverbial phrase (try for solution neutral)**
  - **seat 400 people**
  - **keep time to 1/100 second accuracy**
  - **move 400 people 600 miles in 3 hours**
  - **image quality of a copier**
  - **inspirational or aesthetic value**
- **Goals can be on Form, either at interface or interior**
  - **use GE parts**
  - **Interface to IEEE 488 standard**
  - **don’t use plutonium**
- **Goals can be an overall product/system, its inputs and outputs, with complex connection to form & function**
  - **Buy from a certain supplier**
  - **create new paradigm for this class of products**

Summary - Goals

- **Goal is defined as**
  - **what it planned to be accomplished**
  - **what the designer hopes to achieve or obtain**
- **Expressed in the precise terms of Product Development**
- **Will include goals derived from beneficiary Needs (goals from beneficiaries) i.e. the functional goals**
- **Will also include goals from corporate strategy, regulations, competitive analysis, etc.**
- **Embodied in a statement of goals (requirements ?)**
- **Is defined (in part) by the architect**
- **Exist within, and under the control of the enterprise, and are traded against other attributes**

Assignment for September

For the different upstream influences which flow to goals (in your enterprise) identify:

- **Who is the beneficiary?**
- **How are needs codified and interpreted? By whom?**
- **How are intents incorporated into system goals? By Whom?**
- **Where is the enterprise boundary?**

You will learn more about upstream influences in System Engineering and Marketing.

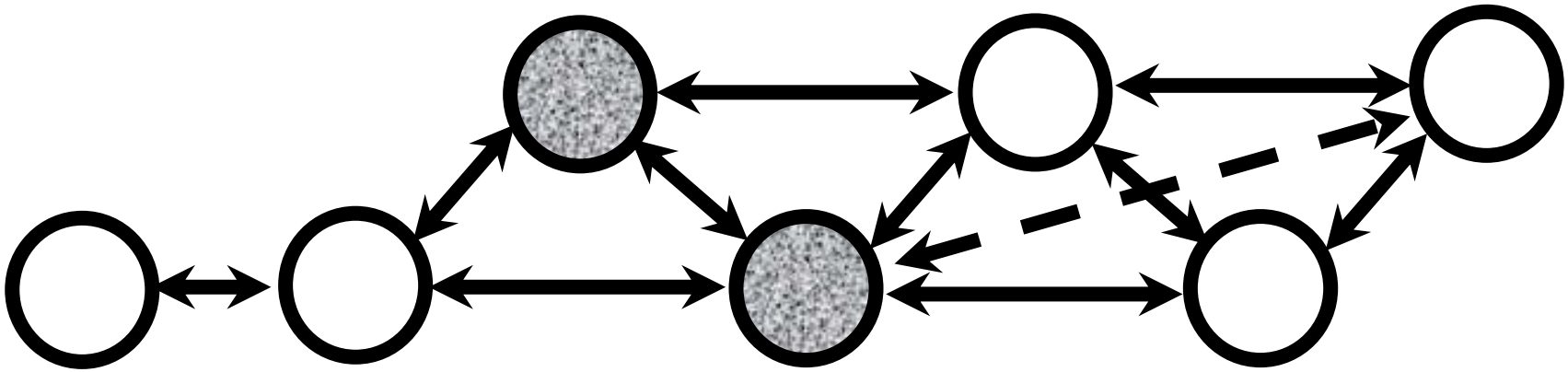
Reverse Engineering from Form to Function and Goal

- **Form is visible, measurable, etc., and can usually be reproduced by good craftsmen (Stradivarius ??)**
- **Functions can usually (or sometimes) be inferred by experts from form by knowledge of underlying physics, logic, common practice and context (operands and other supporting objects)**
- **Quantitative goals can rarely be inferred from form without extensive modeling, and may never be inferable (e.g. gross margin)**

Summary: Upstream Influences

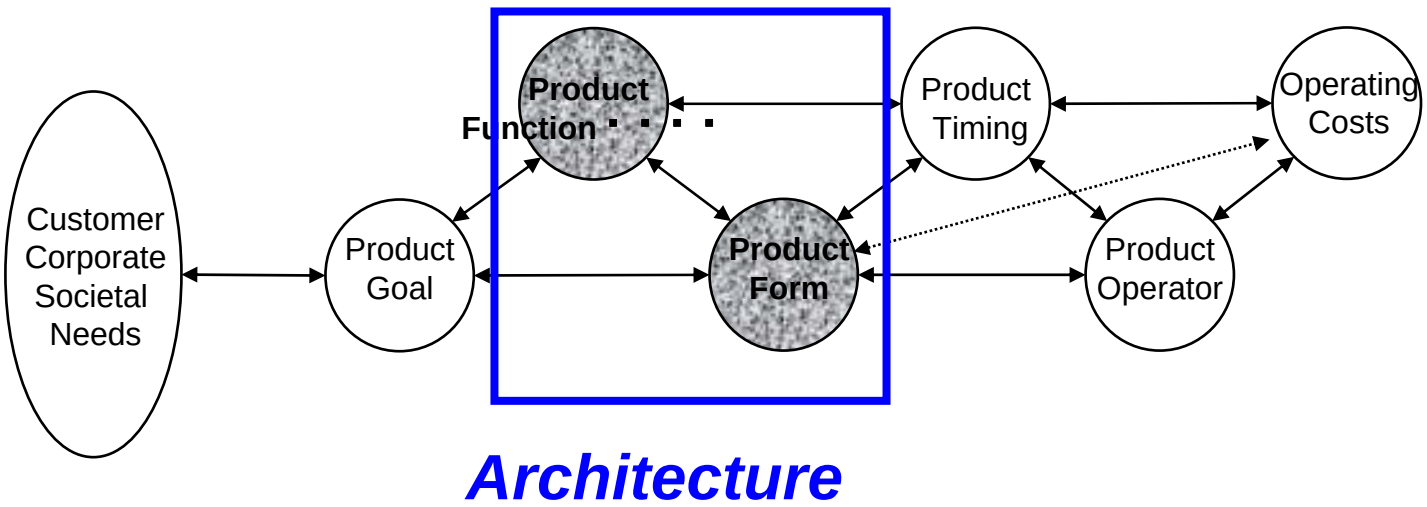
- **Needs exist in the mind of the beneficiary, and are fuzzy and ambiguous and are interpreted (in part) by the architect**
- **Goals are established by interpreting needs, and should be clear and precise. An important goal is the functional intent associated with primary externally delivered value, expressed as a solution neutral function. The architect participates in setting goals**
- **After concept selection, function is defined in the design solution specific domain, ensures goals can be met, and is defined by the architect**
- **Form exists in the physical/informational domain, delivers function, and is defined by the architect**
- **Goals, Function and Form are often traded-off in conceptual design**

Holistic Framework for Product and Operations

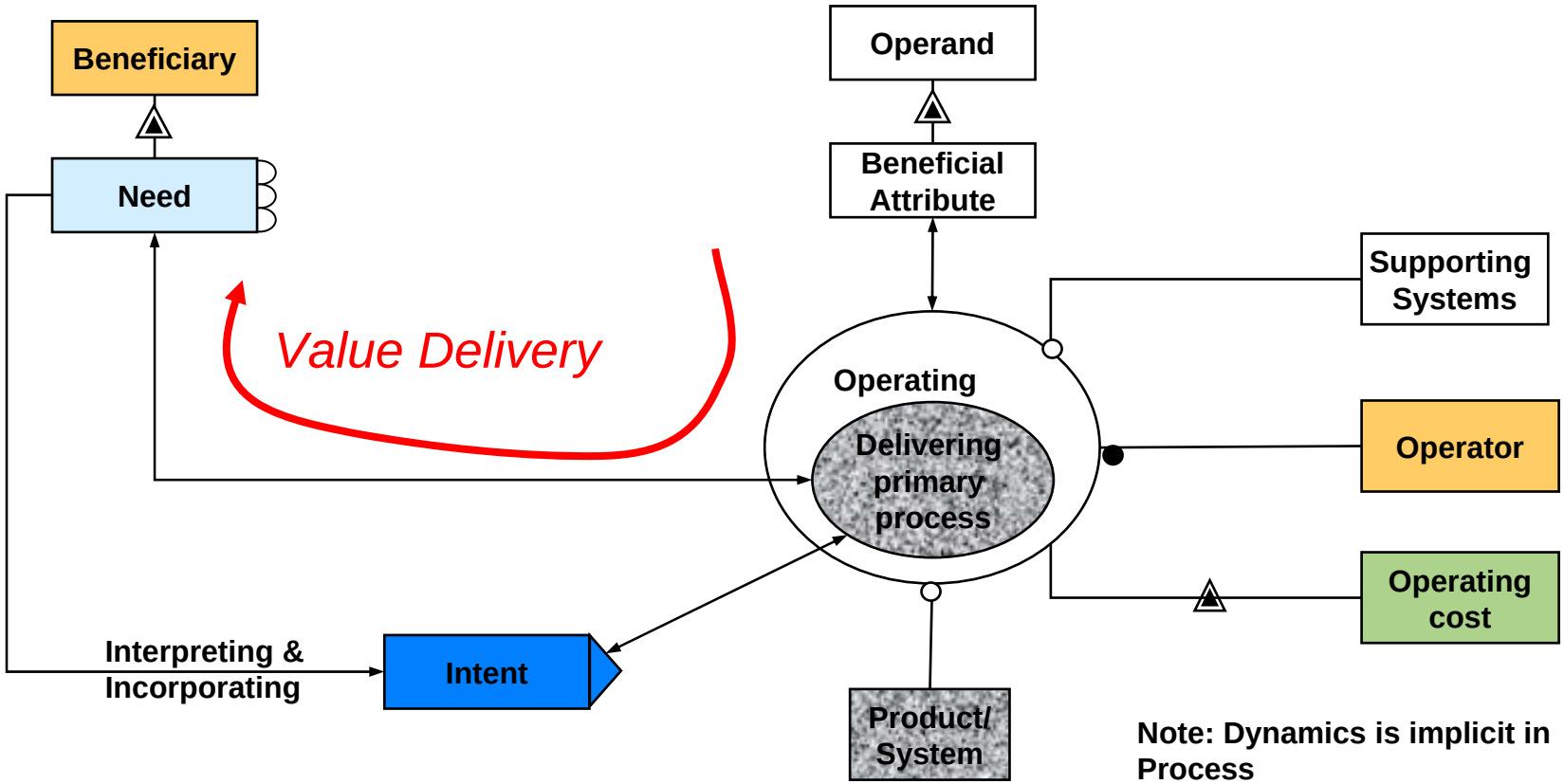


global why	global what	global how	global where	global when	global who	global how much
the system is built	the system accomplishes	the system acts	the elements are	things occur	does them	does it cost
need opportunity	goals performance	function process	form structure	timing dynamics	operator user	cost expense

Summary - Product Attributes

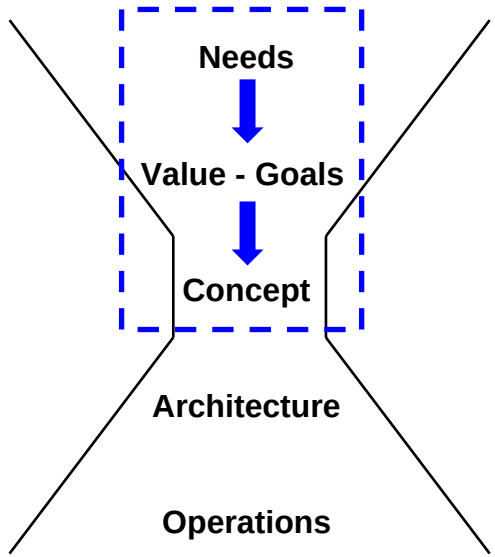


Product Attributes



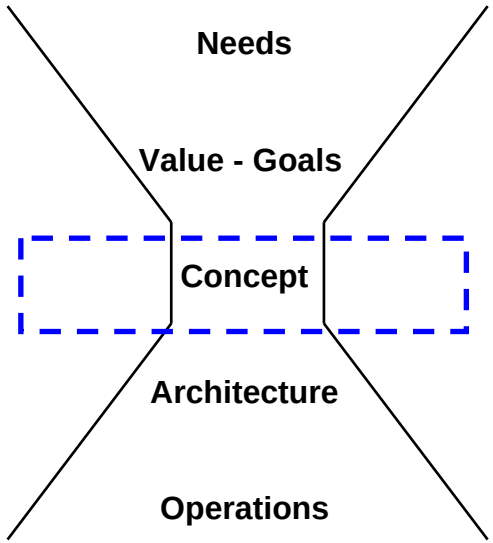
Framework - Needs to Value to Concept

- Who are the beneficiaries? What are their needs?
- What is the value related operand and its states?
- What is a solution neutral statement of the value related transformation - the externally delivered value related function? (solution neutral function)
- What are the solution specific functions which will achieve this transformation? (the function part of concept)
- What are the solution specific abstractions of form that can execute this process?(the form part of concept)
- What are potential multi-functional aspects of the concept process?
- How does value trace to the beneficiaries?



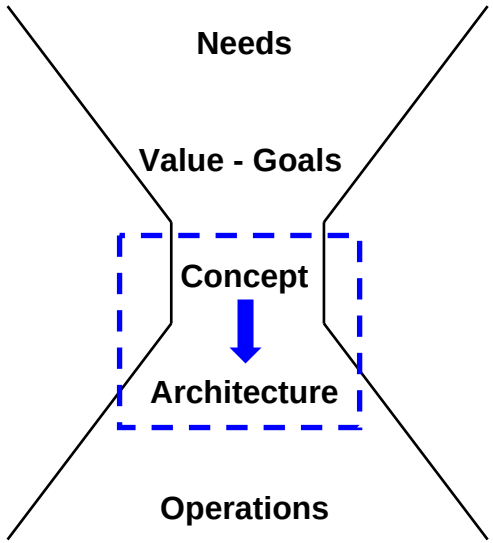
Framework - Concept In Context

- What is the product system?
- What are the supporting systems?
- What is the whole product system?
- What is the use context?
- What are the boundaries?
- What are the interfaces? What are the operands that are passed or shared? Interface process? Interface instrument objects?



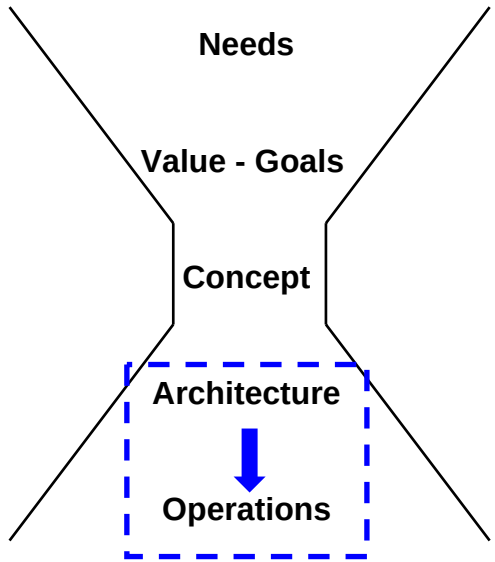
Framework - Concept to Architecture

- What are the principal internal functions? Operands along the way?
- How is the form decomposed into elements? (decomposition of form)
- What is the structure of the elements? How are the internal functions mapped to elements of form?
- How do these combine to produce the emergent externally delivered value related function?
- What other value related external functions are delivered?
- Are there supporting processes and objects?



Framework - Architecture to Operations

- What is the sequence in the process of delivering primary function?
- Are their contingency, emergency or stand alone processes?
- Are there commissioning, decommissioning and maintaining processes?
- Is clock time important to understand in the operations?

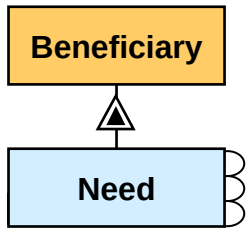


ServeCo

- **ServeCo is a 50 is person engineering design services provider**
- **Typically in the civil/mechanical engineering domain**
- **Typically taking on 3-5 simultaneous projects, lasting about a year**
- **Privately owned by key employees and a few private investors**

Value Questions (1)

- **What is the mission of our enterprise, and the outcomes**



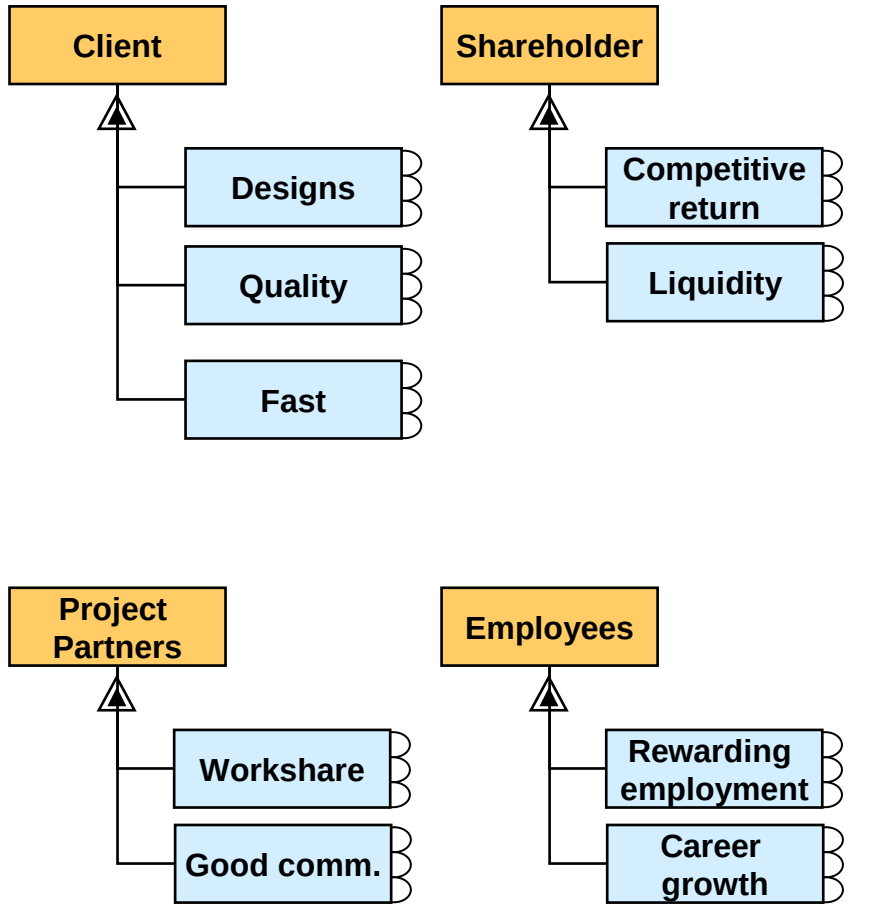
- **Who is the beneficiary?**
- **What is their need(s)?**

Need

▲ Decomposes to
△ Specializes to
◭ Has attribute of

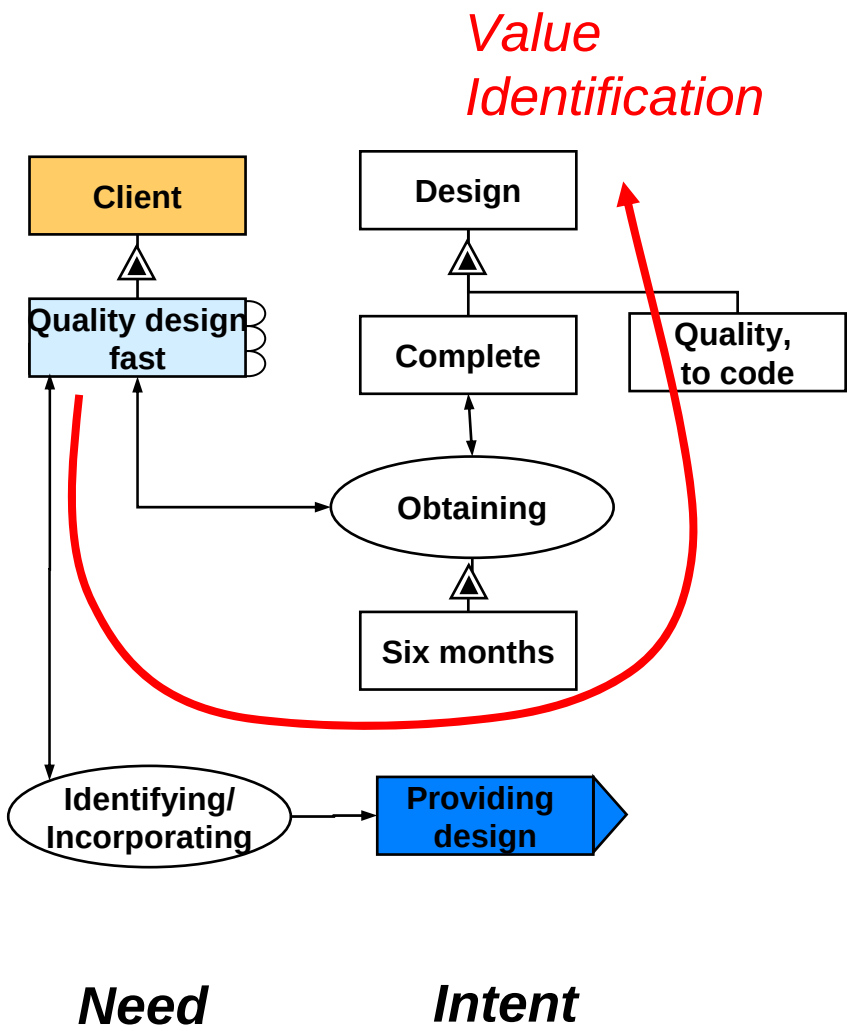
Multiple Beneficiaries - ServeCo

- These are probably key beneficiaries and needs
- There is is a template for many engineering enterprises
- There are other stakeholders - competitors, regulators, suppliers, etc.
- Focus this example only on the client as a beneficiary



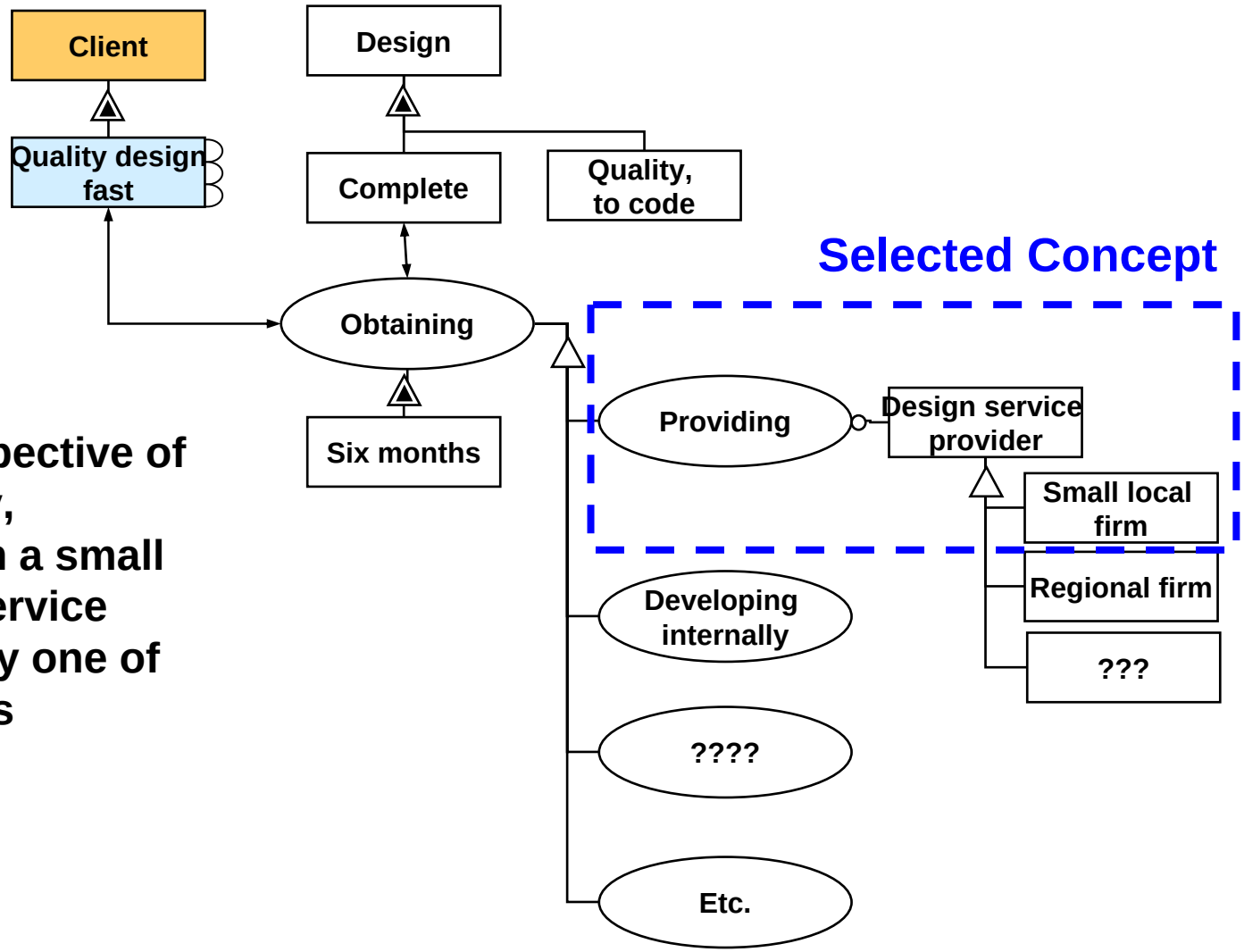
Value Questions (2) - Value Identification

- Who is the beneficiary?
- What is the need?
- What is the value related operand?
- What is the value related attribute?
- What is a solution neutral statement of the value related transformation?
- What are other important attributes of operand and transformation?



Solution neutral statement becomes the intent on function

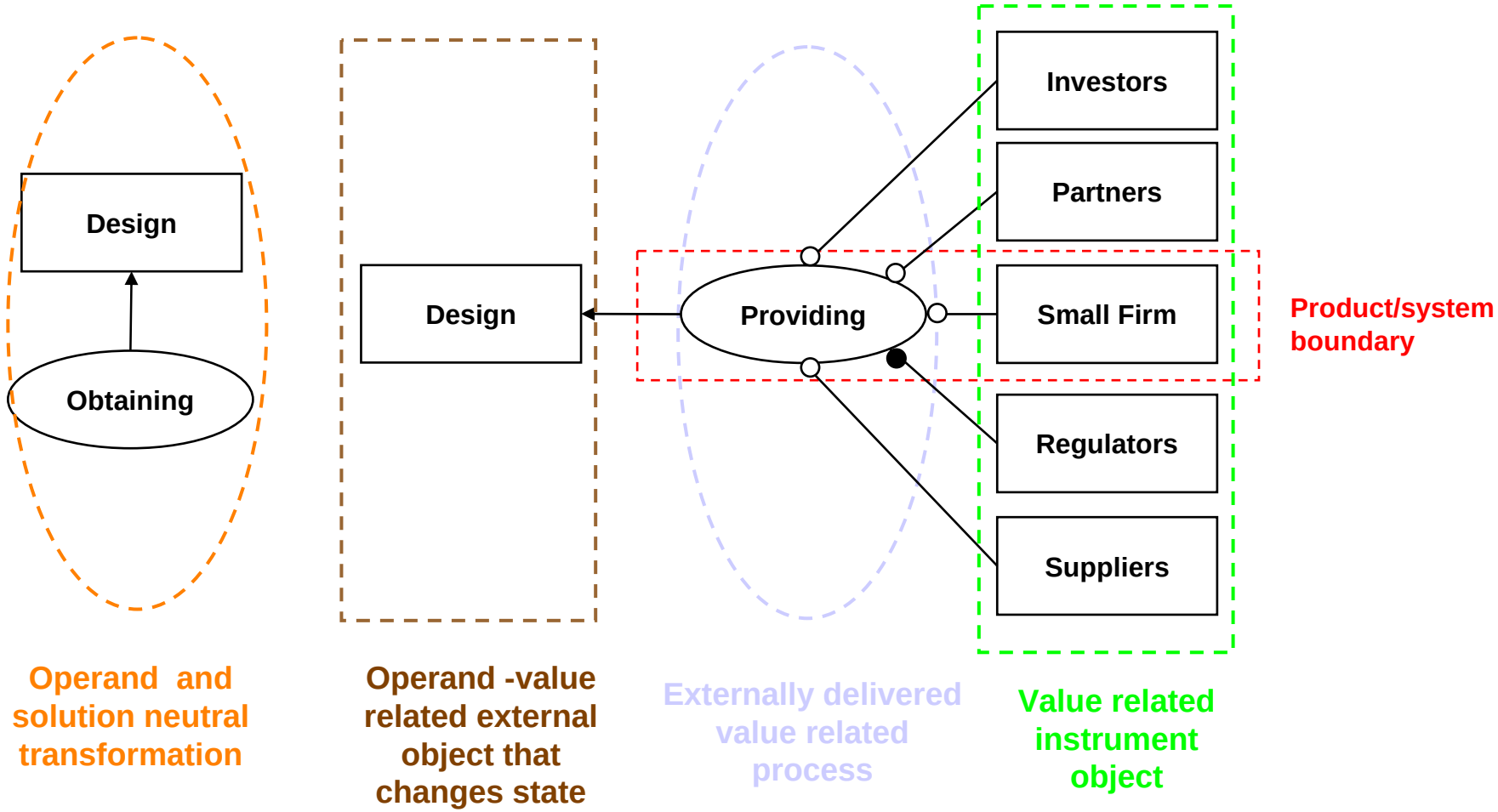
Concepts - ServeCo



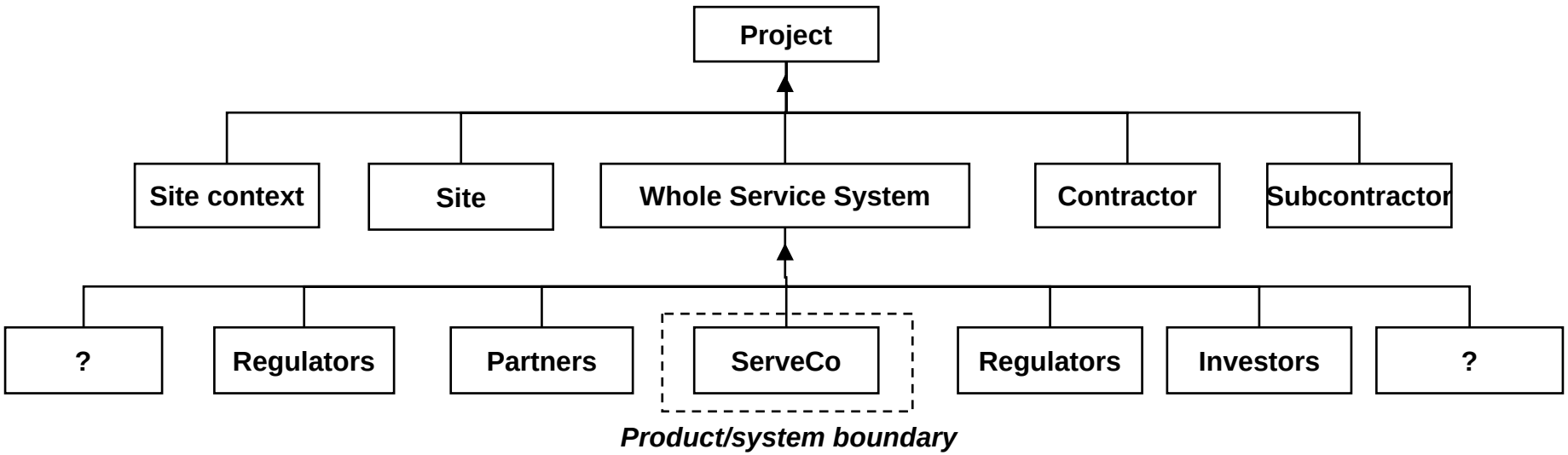
- **From the perspective of the beneficiary, procuring from a small local design service provider is only one of several options**

▲ Decomposes to
△ Specializes to
◭ Has attribute of

Preserving Food Concept - ServeCo



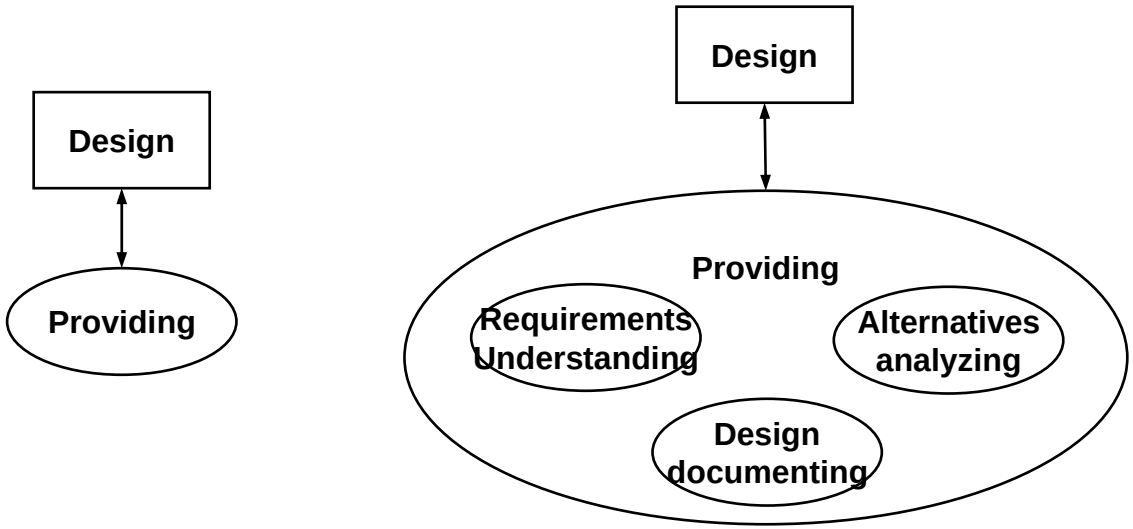
Context - ServeCo



- **What is the whole product system?**
- **What is the usage context in which it fits?**

NB: a complete job would include the architecture of the usage context not just the objects

Multi-function Concepts for ServCo

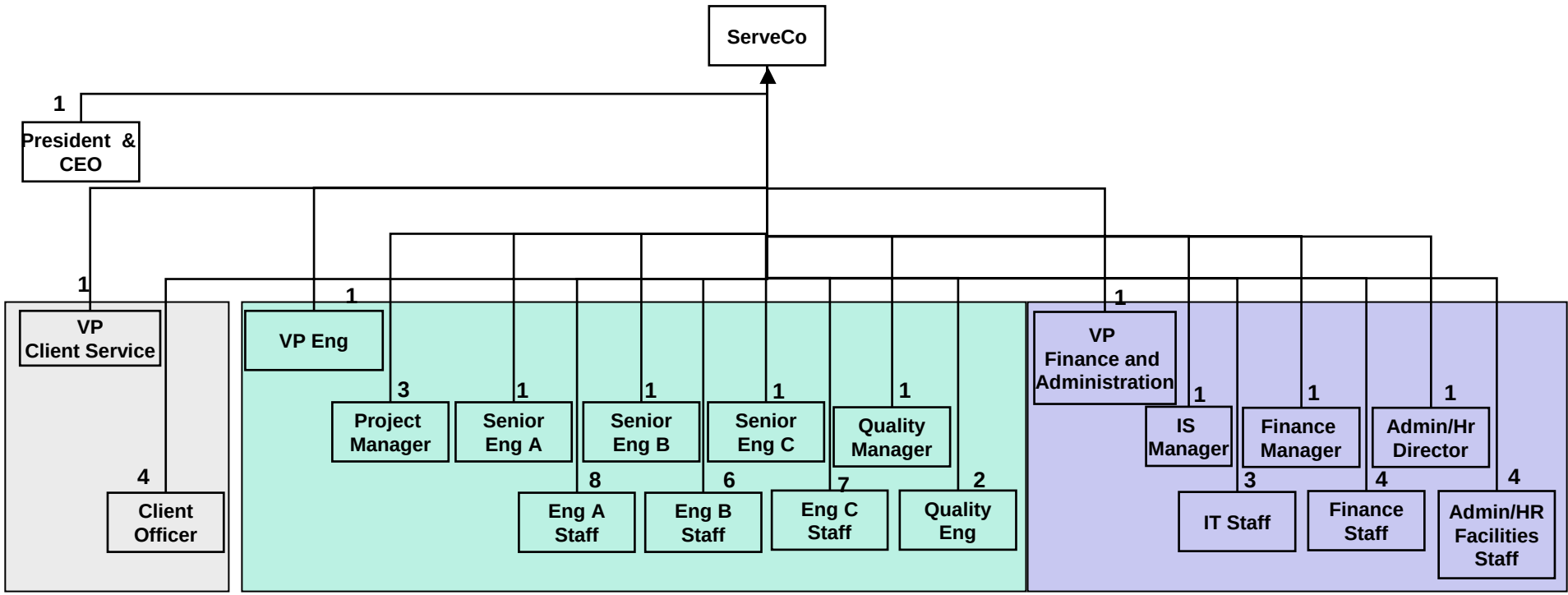


Providing a design implies understanding requirements, analyzing alternatives and documenting the design that emerges

▲ Decomposes to
△ Specializes to
◭ Has attribute of

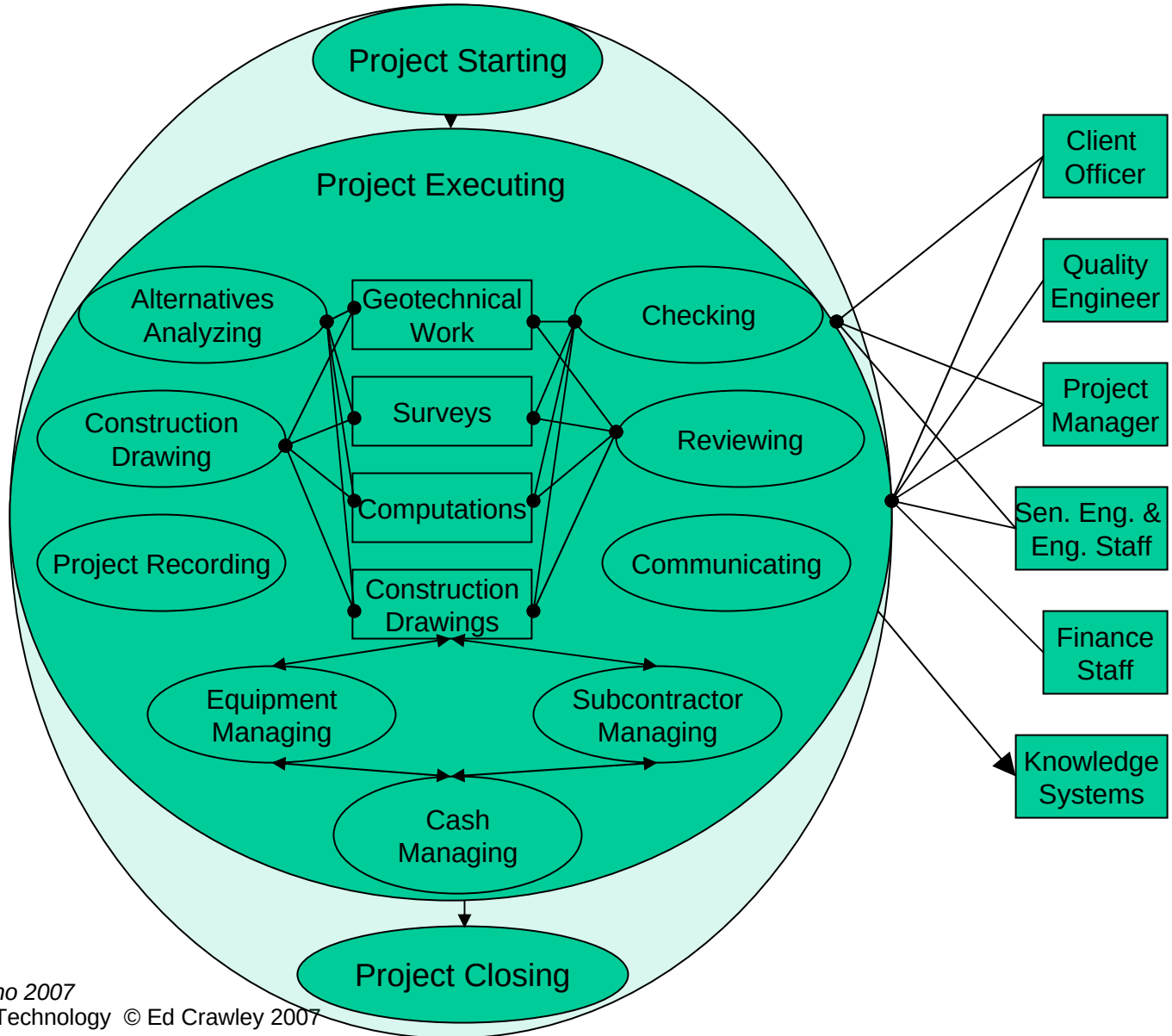
Form of a Medium System - ServeCo

ServeCo is an engineering design service firm
It is staffed by about 50 people in 20 roles

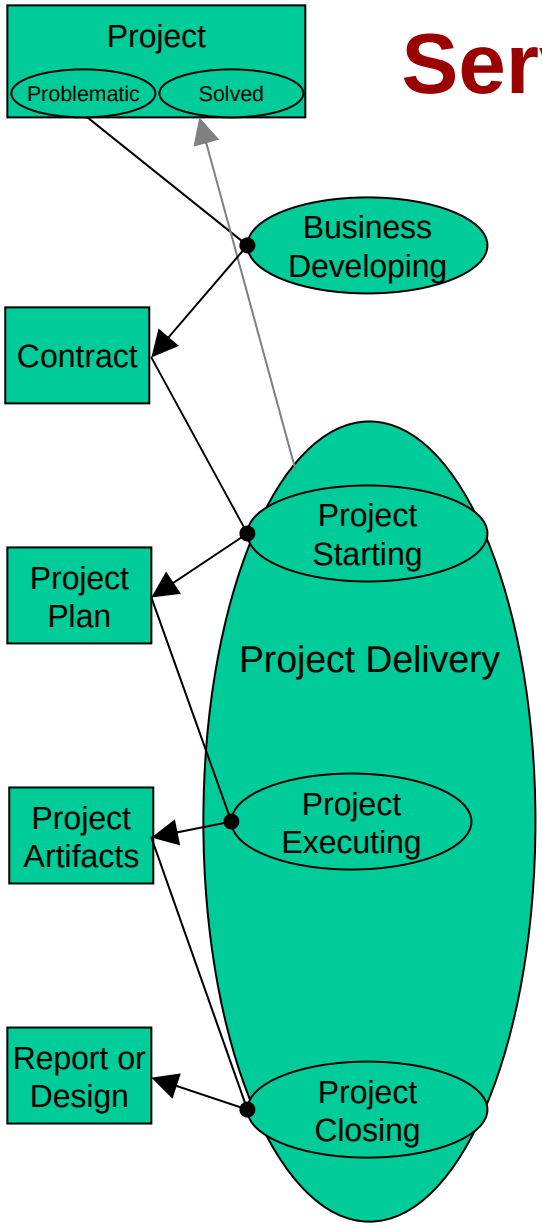


NB: these aggregations are ad hoc and not "correct"

Project Executing Process - Primary Function



ServCo System Operating



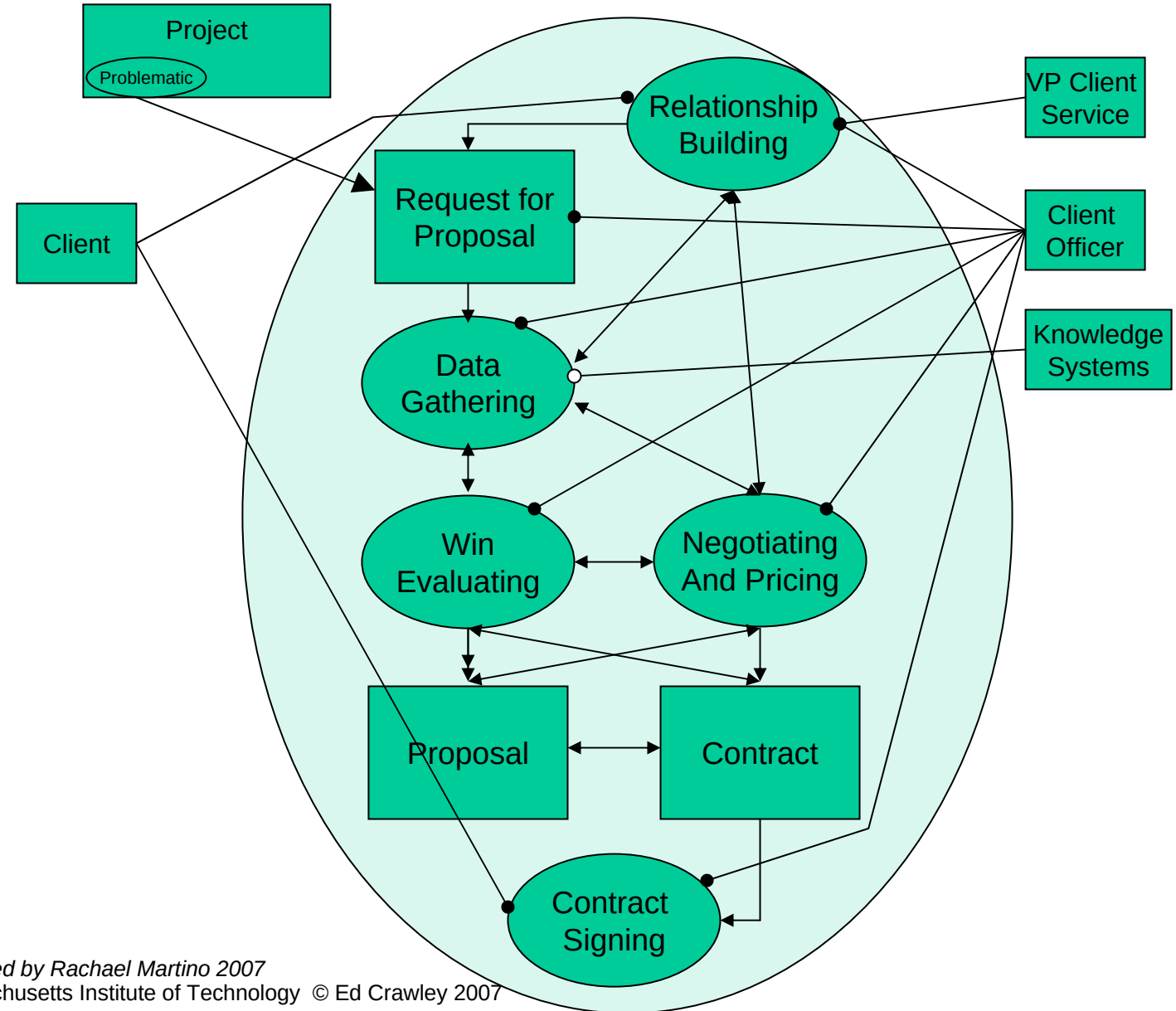
Get Ready

Get Set

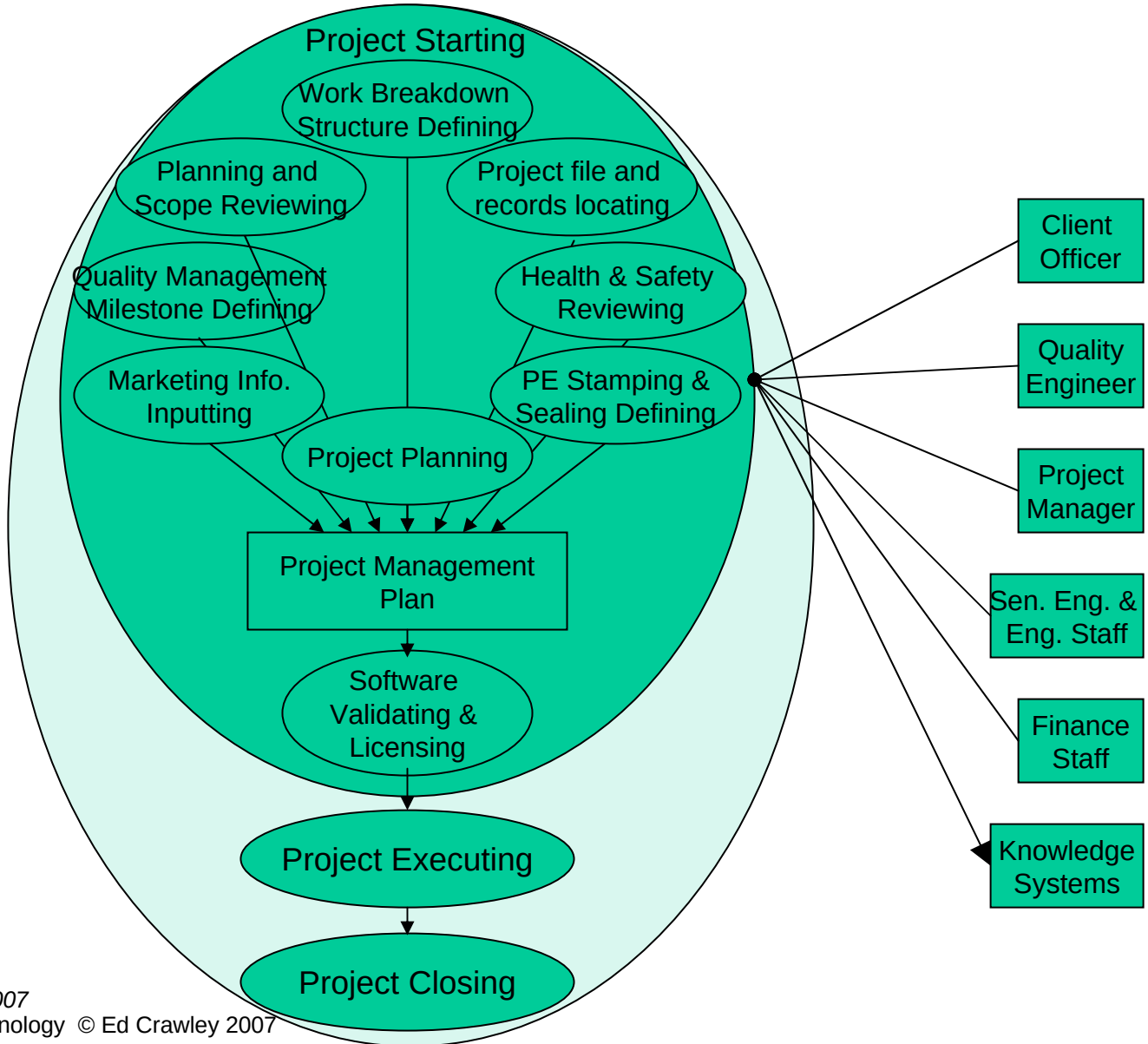
Go

Get Unset

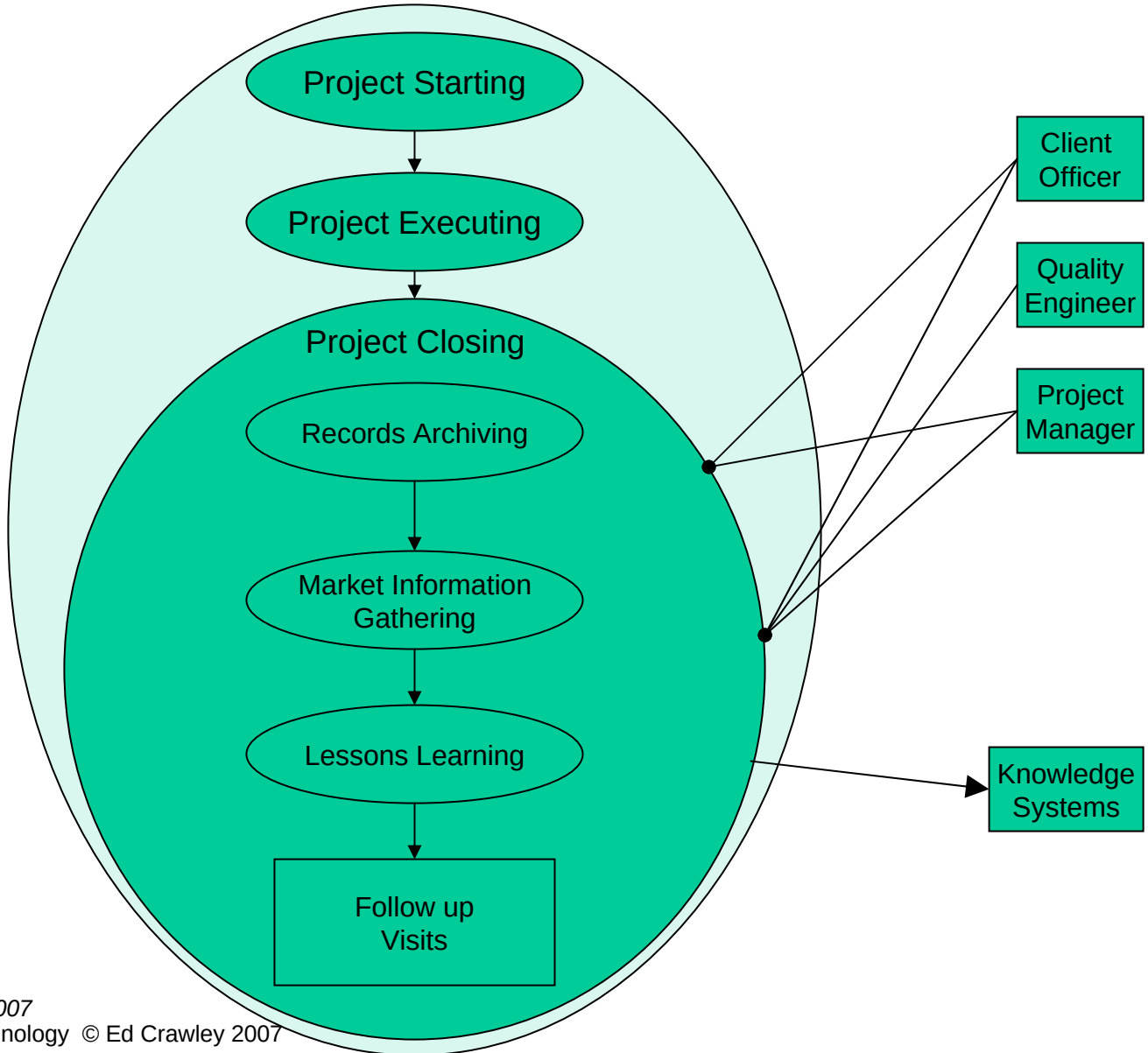
Business Development Process Zoom



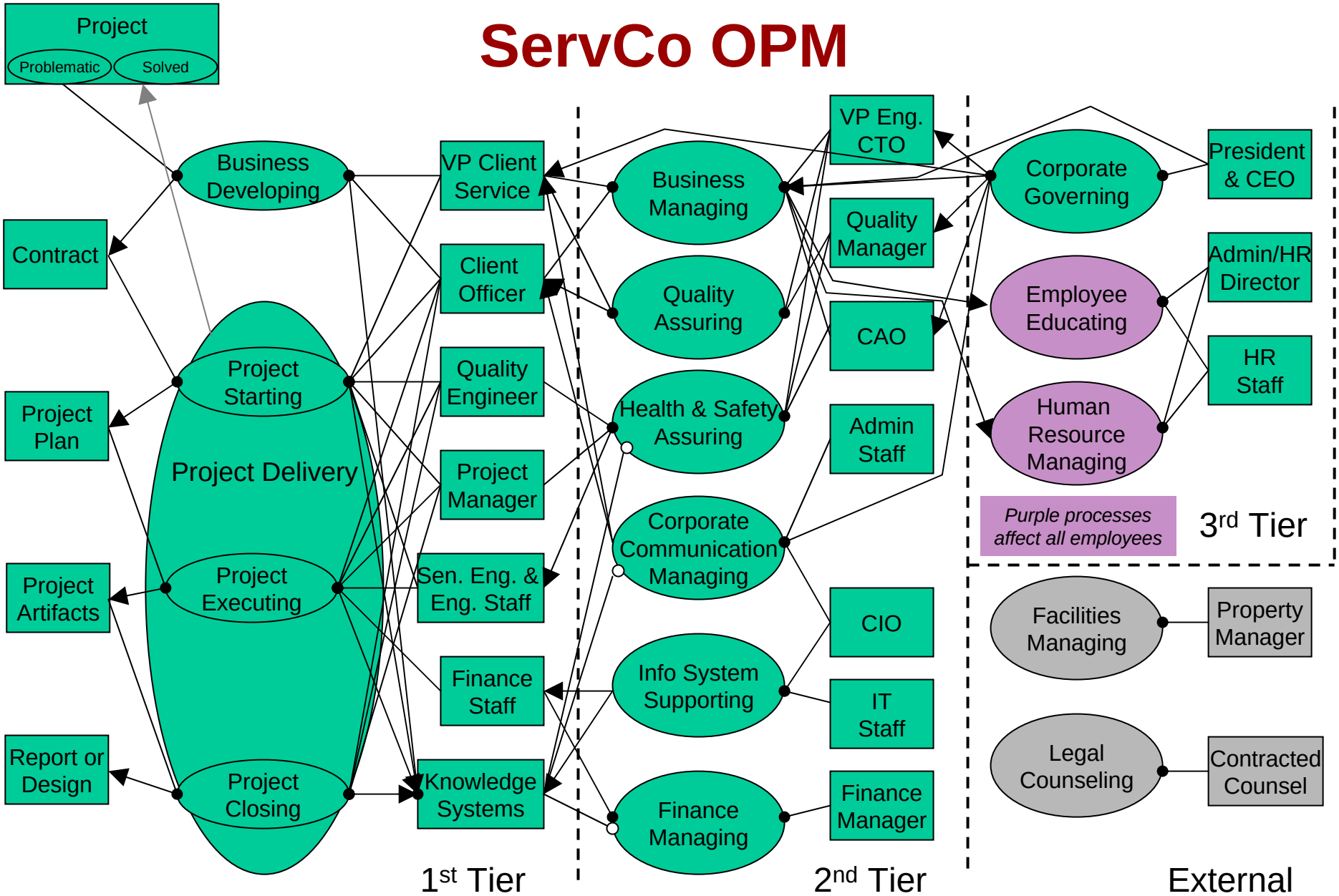
Project Delivery Process Zoom



Project Closing Process Zoom



ServCo OPM



Other Operating Issues?

- **Stand alone ops?**
- **Contingencies?**
- **Emergencies?**
- **Is real clock time important?**

ServeCo Value Delivery