System Architecture

IAP Lecture 3

Ed Crawley

January 19, 2007

Rev 2.0

Today's Topics

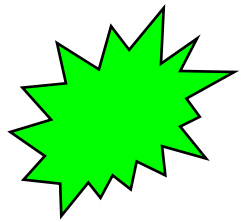
- **Reflection on Function**
- **Concept**
- **Creativity**
- **Architecture**
- **PDP Synthesis**
- **Closure on Definitions (for reference)**

- **Change in style today, I will not talk to every chart, but they are here for study and reference**

Today's Topics

- **Reflection on Function**
  - **Opportunity Set Results**
  - **Zooming**
  - **Solution Neutral Statement of Function**
  - **Expressing Function and Process**
  - **Summary Reflections on Function and Form**
- **Concept**
- **Etc.**

¿ Reflections on Function?

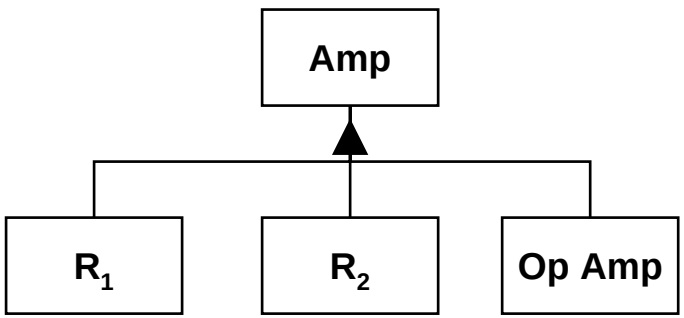


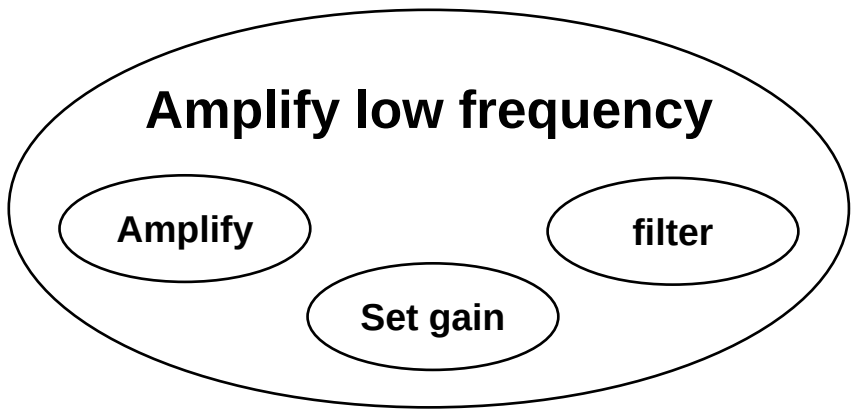
- **What is the value related operand?**
- **What are the value related states that change?**
- **What is the externally delivered function?**
- **Who benefits?**

- **What are the principle internal functions?**
- **How are the internal functions mapped to elements of form?**
- **How do these combine to produce the emergent externally delivered value related function?**

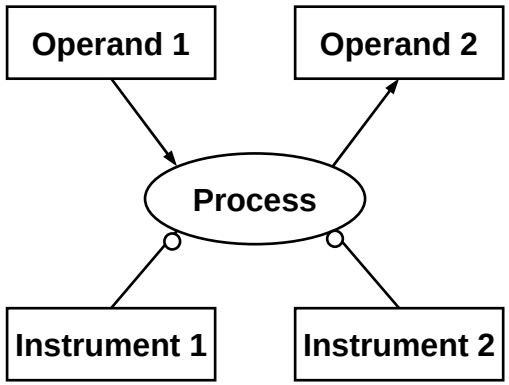
Emergence and Zooming of Processes

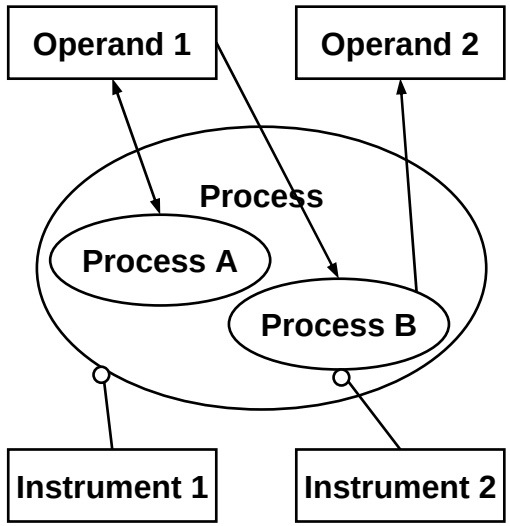
- **A process can be zoomed into sub-processes**
- **A process *emerges* from sub-processes**
- **The process and sub-processes are not linked in any explicit manner, as the system decomposes into elements or the elements aggregate into the whole**
- **Emergence is a powerful feature of systems - elements and sub-processes can come together to cause a process to emerge**





Representation of Zooming





- **Process zooms into sub-processes, processes emerge from sub-processes:**
  - **Object-process arrows can move to sub-processes and be expanded or clarified, or if appropriate can remain attached to the larger process**

Solution Neutral Expression of Function

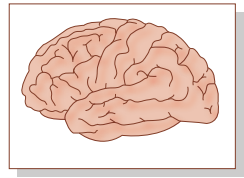
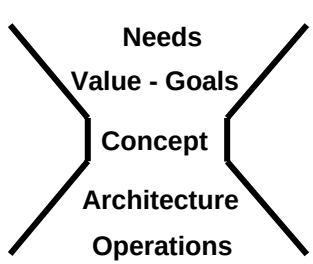


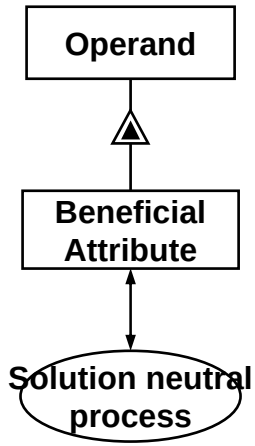
Figure by MIT OCW.

- **Functions should (initially) be expressed as <u>solution neutral statements</u>, having *no solution of specific function or form* either explicit or implicit in the statement**
- **Do this by focusing on the operand(s), and what attribute of the operand(s) you wish to change**
- **Solution neutral statements:** **Not:**
  - **Cargo transporting** **Trucking**
  - **Message communicating** **Phoning**
  - **Food heating** **Broiling**
  - **Idea creating** **Brainstorming**
- **This will focus on the creation of *value*!**
- **This will also foster *creativity*, by allowing a wider range of possible solutions to be imagined!**

Solution Neutral Statement of Function - Questions?

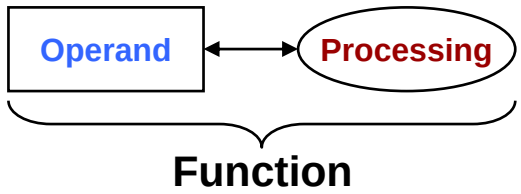


- What is the value related operand?
- What are the value related changes in an attribute - the beneficial attribute?
- What is the solution neutral transformation of these attributes?
- This is the sought after "solution neutral function"



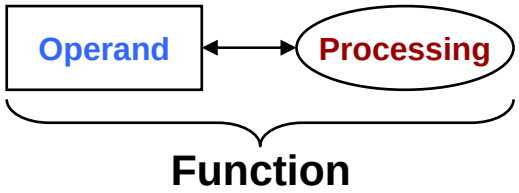
▲ Decomposes to
△ Has attribute of

Exit Row Passenger Functions?



- **In the event of an emergency, a passenger seated in an exit row should:**
  - **Locate the emergency exit**
  - **Recognize the emergency exit opening mechanism**
  - **Comprehend the instructions for operating**
  - **Assess whether opening will in crease the hazards**
  - **Follow crew instructions**
  - **Stow the emergency exit door**
  - **Assess the condition of the escape slide**
  - **Pass expeditiously through the exit**
  - **Assess, select and follow a safe path away from the exit**

Expressing Process - Limited Syntax



- **The assertion is that all process can be reduced to one of a limited set**
- **Trying this forces you to consider if the step is truly a process, and of what limited type it might be**
- **One useful set, developed by Krumhauer, is:**
  - **Channel (transport in place)**
  - **Store (for a period of time)**
  - **Change (in nature or type)**
  - **Vary (in magnitude)**
  - **Connect (multiple inputs to single outputs, or vice versa)**
- **Try it on your next opportunity set, and see if it works**

Ref: Pahl and Beitz, after Krumhauer

Fundamental Processes

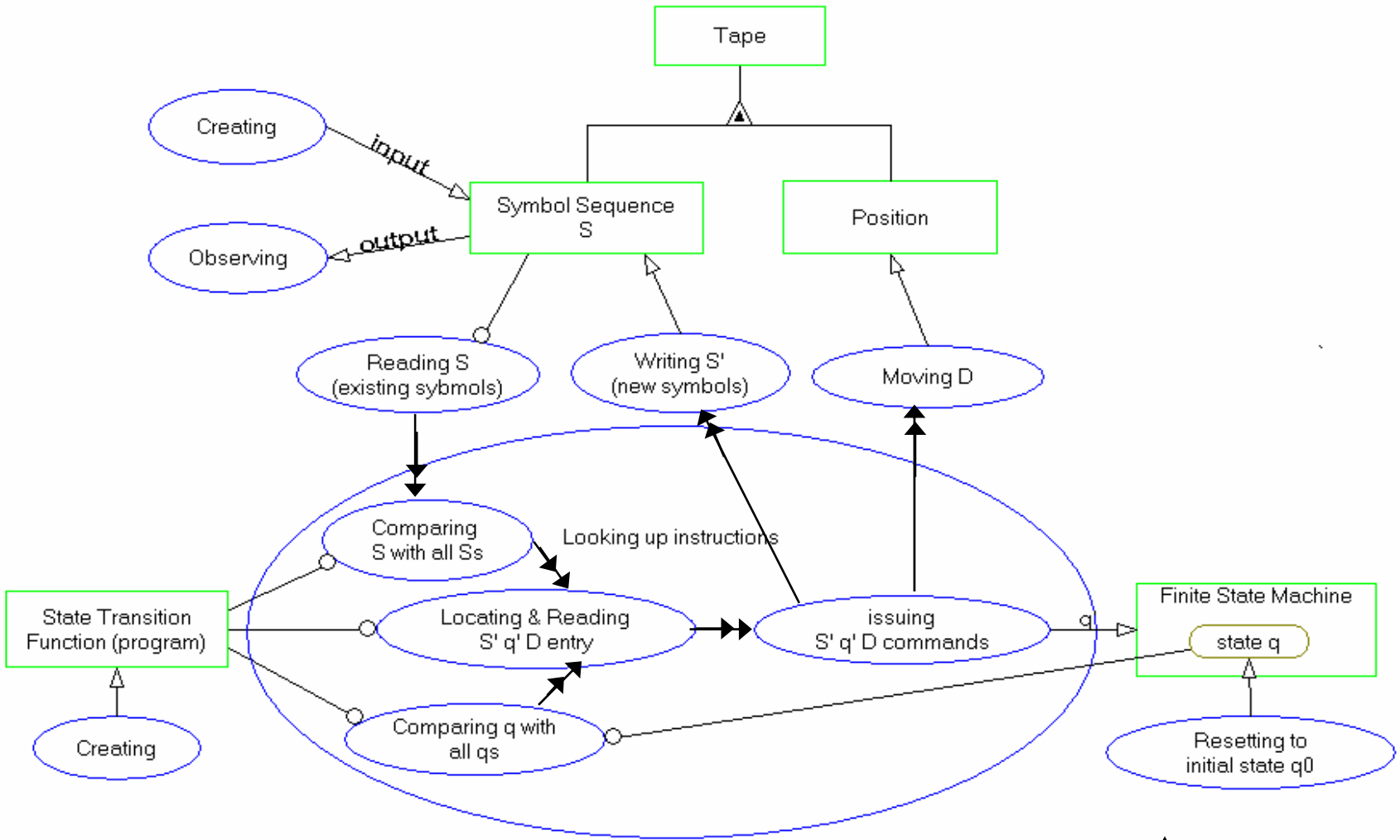
A la Crawley

- **Create (and Destroy)**
- **Transport**
  - **In place - from A to B, or to "spatial storage" and recover from "storage"**
  - **In time – only delays allowed since time is causal "temporal storage"**
- **Transform**
  - **In type or form**
  - **In quantity – magnitude for continuous attributes, number for discrete artefacts**
- **Compare**
  - **Any of the place, time, type or quantity [not sure it is independent of Transform]**

Fundamental Process Frameworks

Dori	**Create, Destroy**	**Transform**					**-**
Crawley	**Create, Destroy**	**Transport**		**Transform**			**Compare**
		Place	**Time (delay)**	**Type/ Form**	**Quantity**		
					Magnitude (continuous)	**Number (discrete)**	
Pahl & Beitz	**-**	**Place (channel)**	**Time (store)**	**Type (change)**	**Magnitude (vary)**	**Number (connect)**	
Turing	**Create**	**Move**	**Store**	**Read, write**	**-**	**Write**	**Look up (compare and locate)**
Bool	**-**	**-**		**-**	**-**	**And, Or**	**(Equivalence, If)**

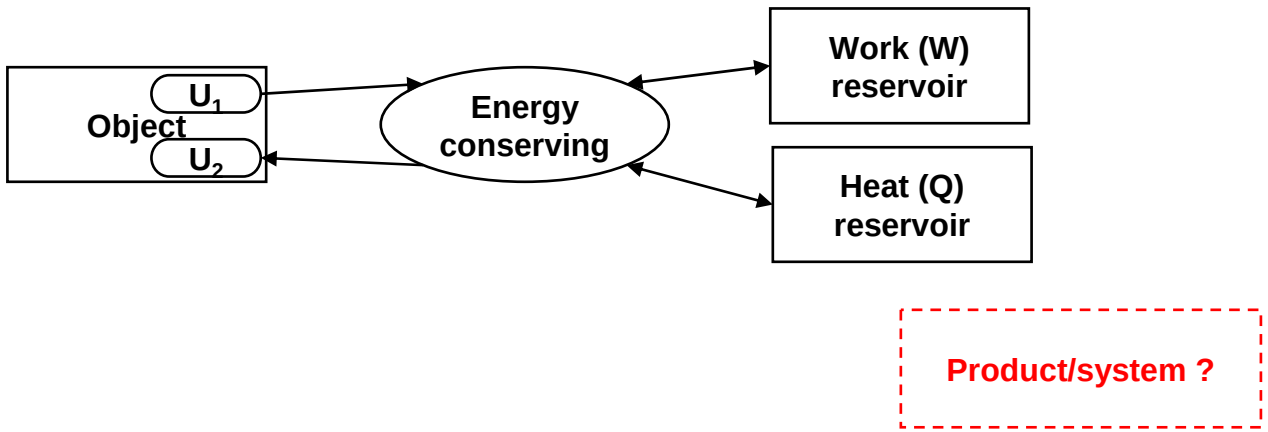
Turing Machine



▲ Decomposes to

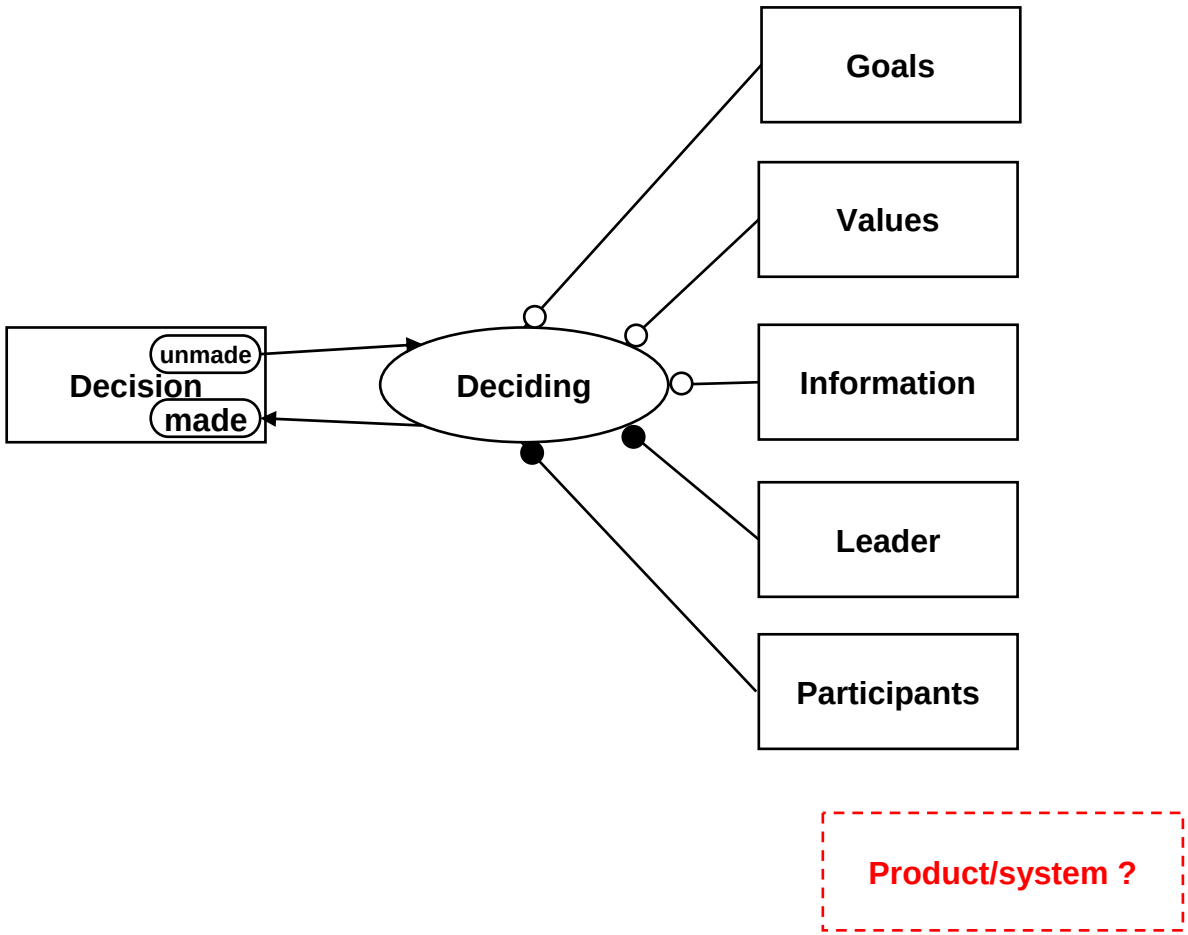
◬ Has attribute of

OPM of a Fundamental Physical Process



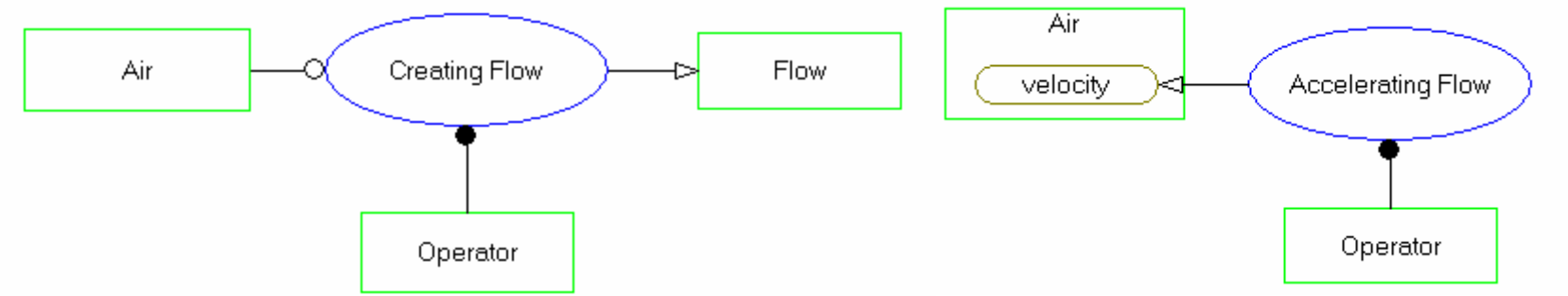
Generally, in detailed physical systems, equations represent processes, and variable represent the state of the objects

OPM of a Social System



Representing “Creation”

- **For informational objects, which are abstractions, the process creating can readily be used (e.g. writing a poem, creating a theory)**
- **Care must be taking in using *creating* for physical objects**
  - **Often can use creating with an abstraction**
  - **Or alternatively transformation with a more concrete description**



- **Is creating *ever* strictly appropriate for a physical system??**

Summary - Function

- **Function is the activity, operations, transformations that create or contribute to performance - it is operand + process**
- **Function is enabled by form, and emerges as form is assembled**
- **Externally function delivered to the operand is linked to the benefit of a product/system**
- **The process part of function can be zoomed, and is potentially expressible with a limited syntax**
- **Function is a system attribute, and initially expressed by the architect in solution neutral statement**

Informational vs. Physical Processes and Objects - “Duality”

- Things (objects and processes) can be physical or informational
- Physical things deal with matter and energy, are “tangible”, and obey the “laws of nature”. Physical objects have mass (particle/wave), and occupy coordinates in space and time
- Informational things are not bound by the laws of nature. An informational object is a piece of information in the abstract (e.g. a database, an idea, a rule, a command). An informational process is some transformation of information (e.g. reading, storing, learning, creating, etc.)
- Informational things always have a physical manifestation somewhere (e.g. design → print, idea → neurons, rule → law book)
- Physical things implicitly contain the information necessary to describe them (but not derive them)
- Processes can have records, plans, etc which are also informational objects

Interfaces Have Form and Function

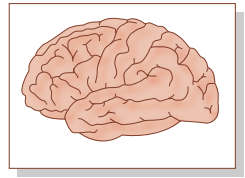
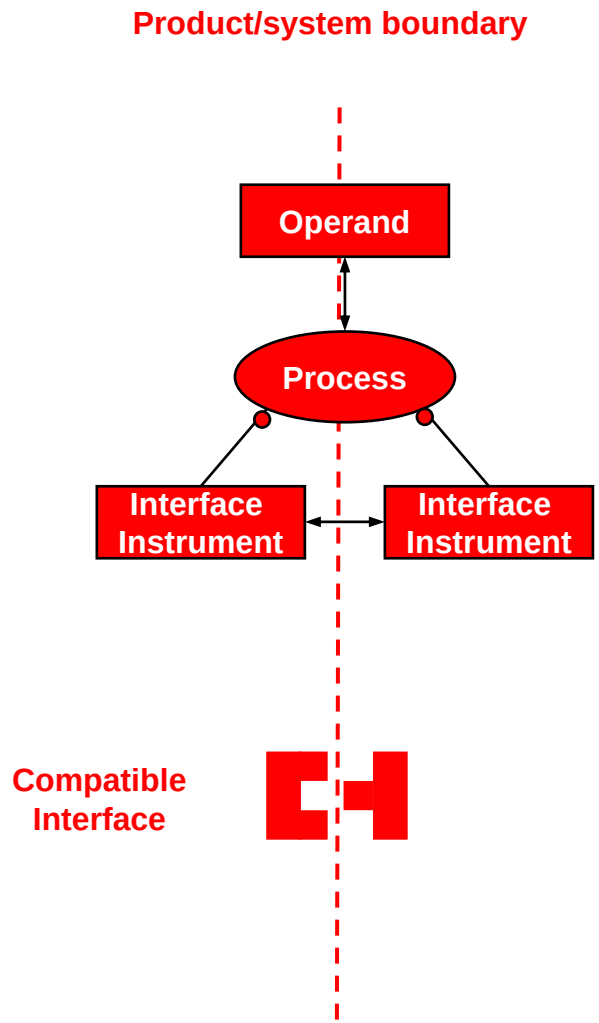
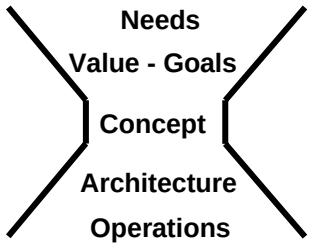


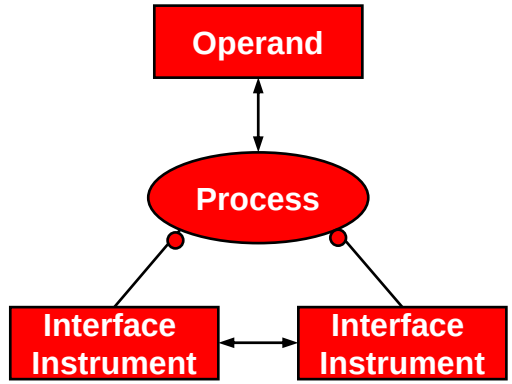
Figure by MIT OCW.

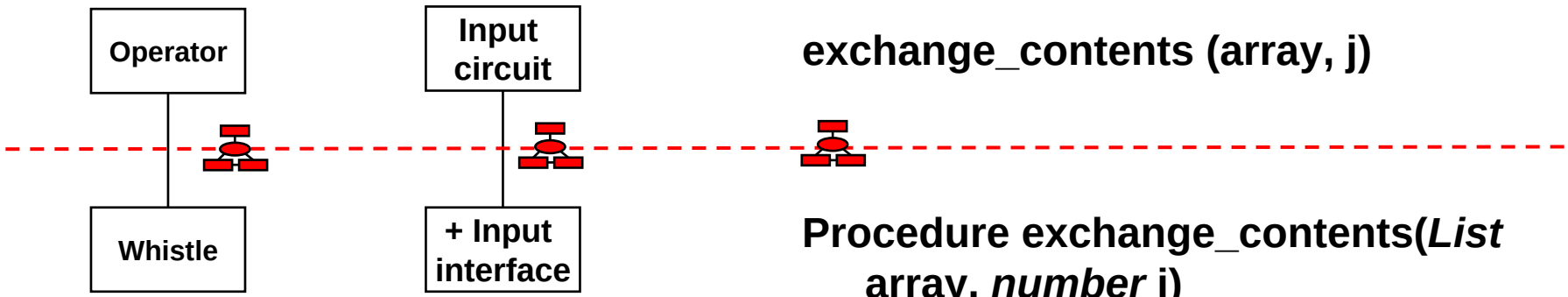
- **The structure usually indicates the *existence* of an interface (more about this next time)**
- **At the interface:**
  - **Form has some structural relationship - usually compatible**
  - **A function is performed - usually the process is the same or the complement**
  - **The operand is the same**



Interfaces - Questions



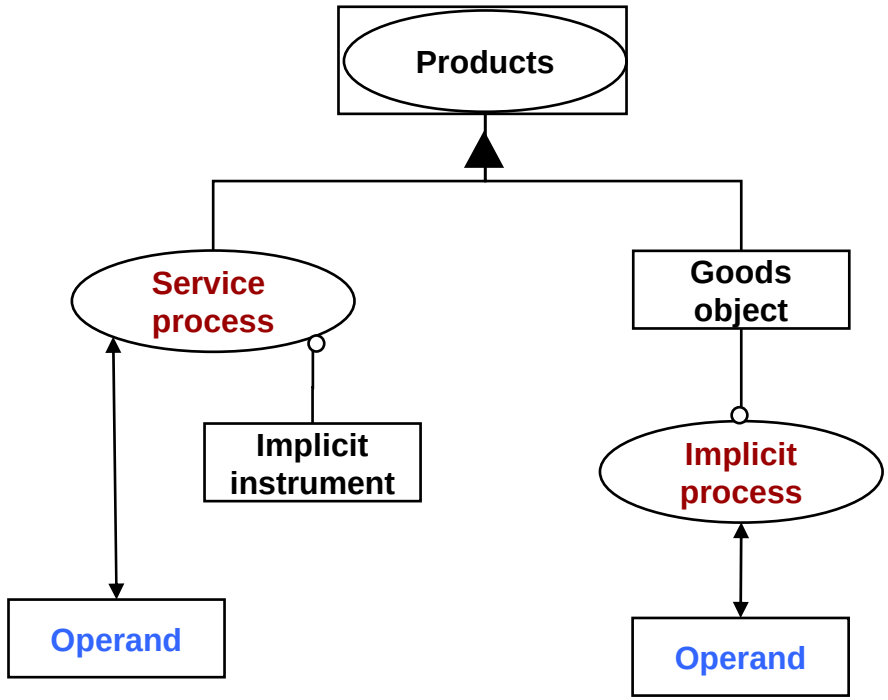




- **What is the operand(s) that is passed or shared?**
- **What is the process(es) at the interface?**
- **What are the instrument objects of the interface, and how are they related (identical, compatible)?**

Goods and Services

- Goods are objects
- Services are processes
- There is always an operand
- With every product good object, there is an implicit process which is linked to value
- With every product service process, there is always an implicit instrument object

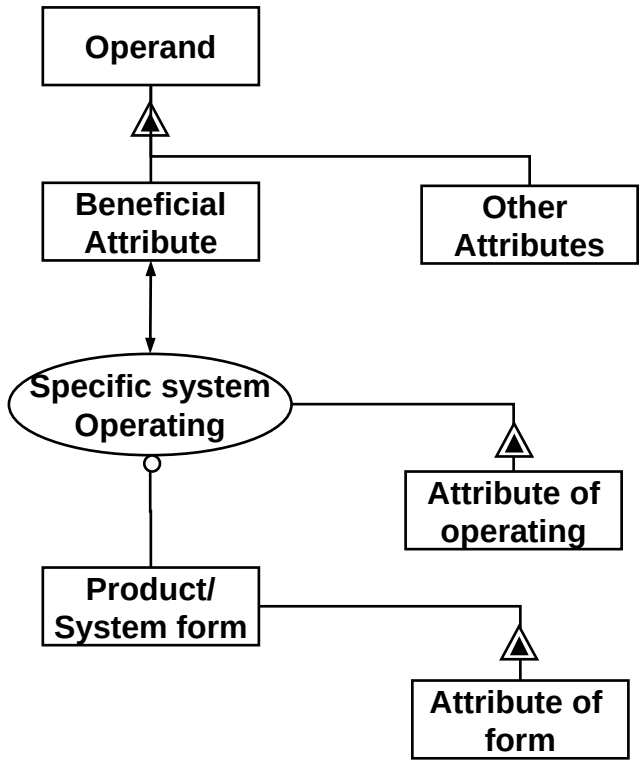


Product/systems always come in object-process-objects, and value is always linked to process acting on operand

▲ Decomposes to
△ Has attribute of

Objects and Processes in Natural Language

- **Objects are nouns: subjects (agents and instruments) and predicates (operands)**
- **Processes are verbs**
- **All human languages are in one of two patterns: NNV or NVN**
- **Read down for passive voice, up for active**



The combination of Operands, Processes and Instrument Objects, together with attributes, can represent human language, and therefore the systems that can be described with human language

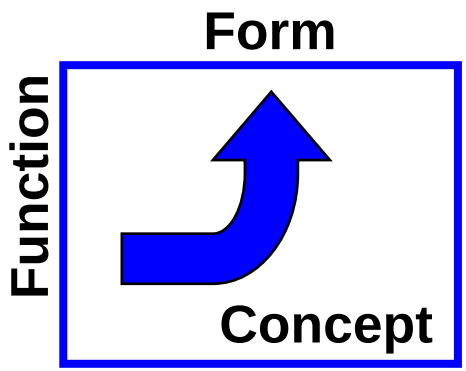
Summary - Form and Function

Form:	Function:
What a system is	What a system does
Objects + Structure	Operands + Processes
Aggregates (and Decomposes)	Emerges (and Zooms)
Source of Cost	Source of External Benefit
Specified at an interface	Specified at an interface
Enables function	Requires instrument form

Form and Function are completely different ideas -
Engineers tend to focus on the concrete, the form, and hence not emphasize the link to value provided by function

Architecture

- **Consists of:**
  - **Function**
  - **Related by <u>Concept</u>**
  - **To Form**



Form - Defined

- **The physical/informational embodiment which *exists*, or has the potential to exist**
- **Is what the system "is"**
- **The *sum* of the elements, which are segments (of the whole of) the form**
- **The structure of form - the formal relationships among the elements**
- **Is a system/product attribute**

Form is Elements + Structure

Function - Defined

- **The activities, operations and transformations that cause, create or contribute to performance (i.e. meeting goals)**
- **The actions for which a thing exists or is employed**
- **Is a product/system attribute**

Form is Operand + Process

Concept - Definition

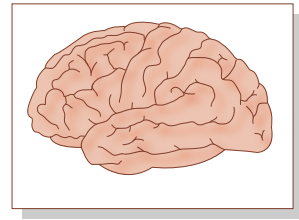
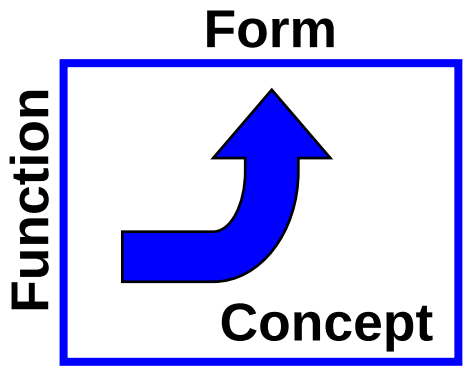


Figure by MIT OCW.

- **A product or system vision, idea, notion or mental image which maps Function to Form**
- **Embodies *principle of operation***
- **Includes an *abstraction of form***
- **Concept rationalizes the structure of the architecture (Imrich)**
- **Establishes the solution-specific vocabulary - it is the beginning of the architecture**



Concept is not a product/system attribute, but a mapping

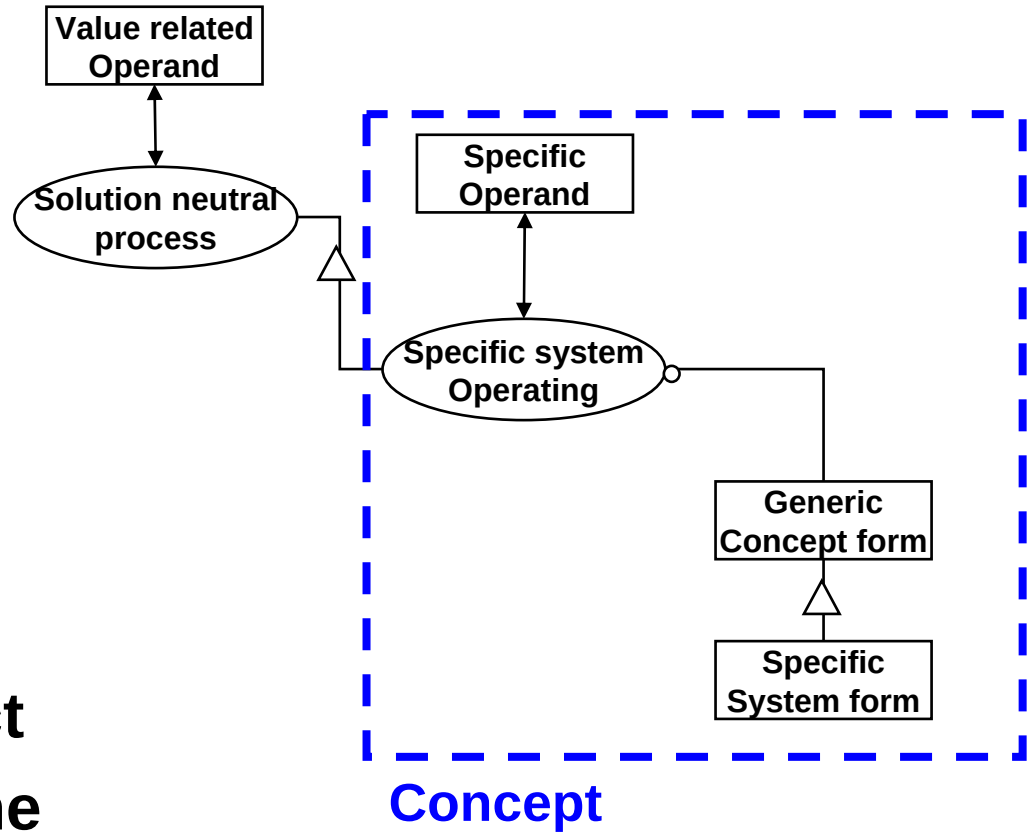
Concept - Described

- **Is *created* by the architect**
- **Must allow for execution of all functions**
- **Establishes the design parameter**
- **Implicitly represents a level of technology**

Managing and focusing *creativity* to create the concept is a main role of the architect during the architecting process

Concept - Formal Definition

- **The combination of:**
  - **The specific system operating process and operand**
  - **The specific system form object**
  - **Related through the generic form object**



▲ Decomposes to
△ Specializes to
◬ Has attribute of

Specialization

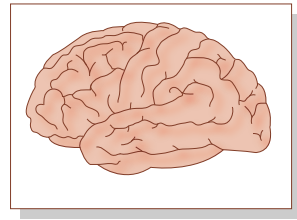
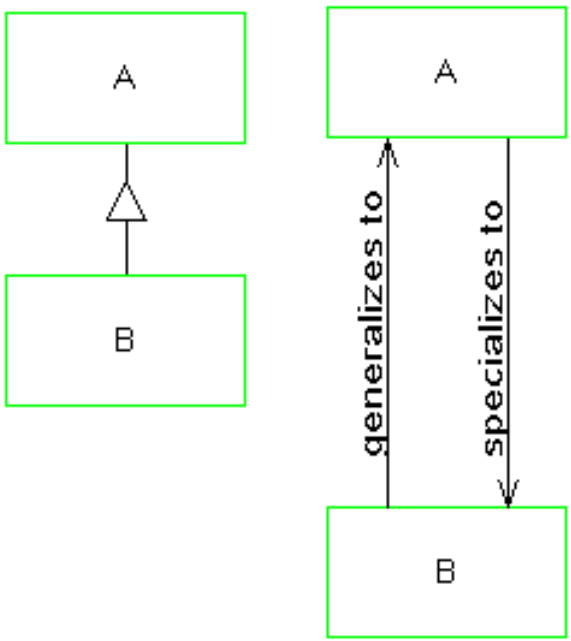
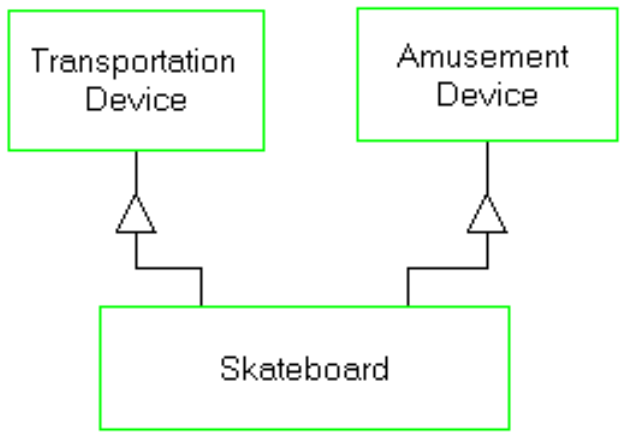


Figure by MIT OCW.

- **Specialization/Generalization**
  - **The relationship between a general object and its specialized forms**

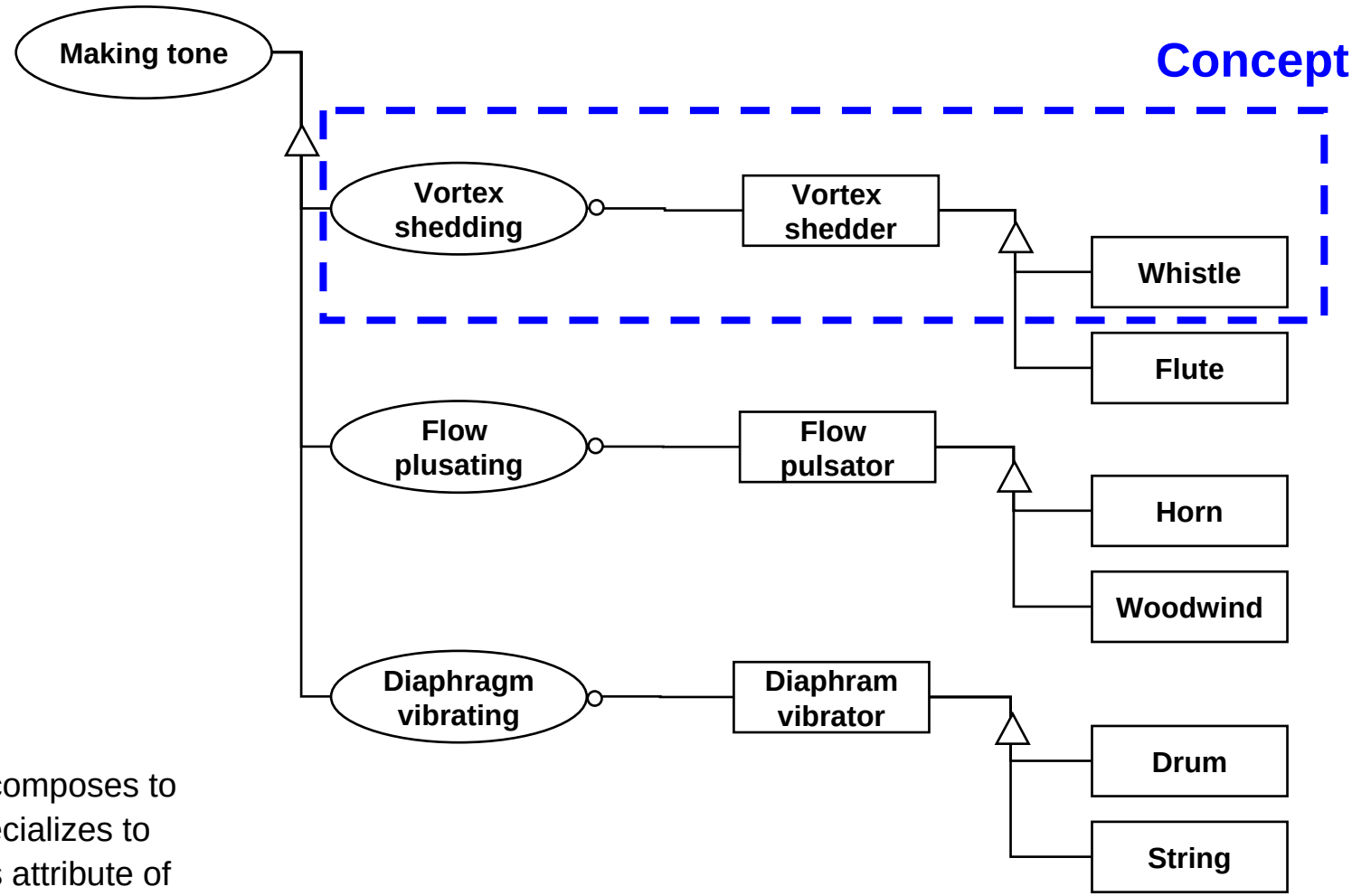




Concepts - Making Tone

Solution neutral statement of function	Solution specific function	Solution specific abstraction of form
Making a tone	Vortex shedding and amplifying	Whistle, Flute
	Air flow pulsating and amplifying	Horns, Woodwinds
	Diaphragm shaking and amplifying	Drums, Strings
	?	?

Concepts - Making Tone



▲ Decomposes to
△ Specializes to
◬ Has attribute of

Concepts - Transporting?

Transporting Concepts

- walk
- ride animal
- wagon
- balloon
- train
- bicycle
- automobile/truck
- airplane
- helicopter
- rocket
- ??

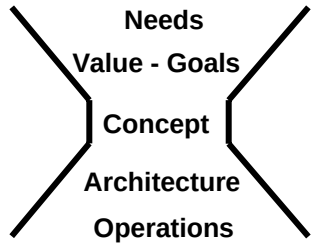
- What is the solution neutral statement of function?
- What is the solution specific function?
- What is the solution specific abstraction of form?

- All are present, but may be implicit

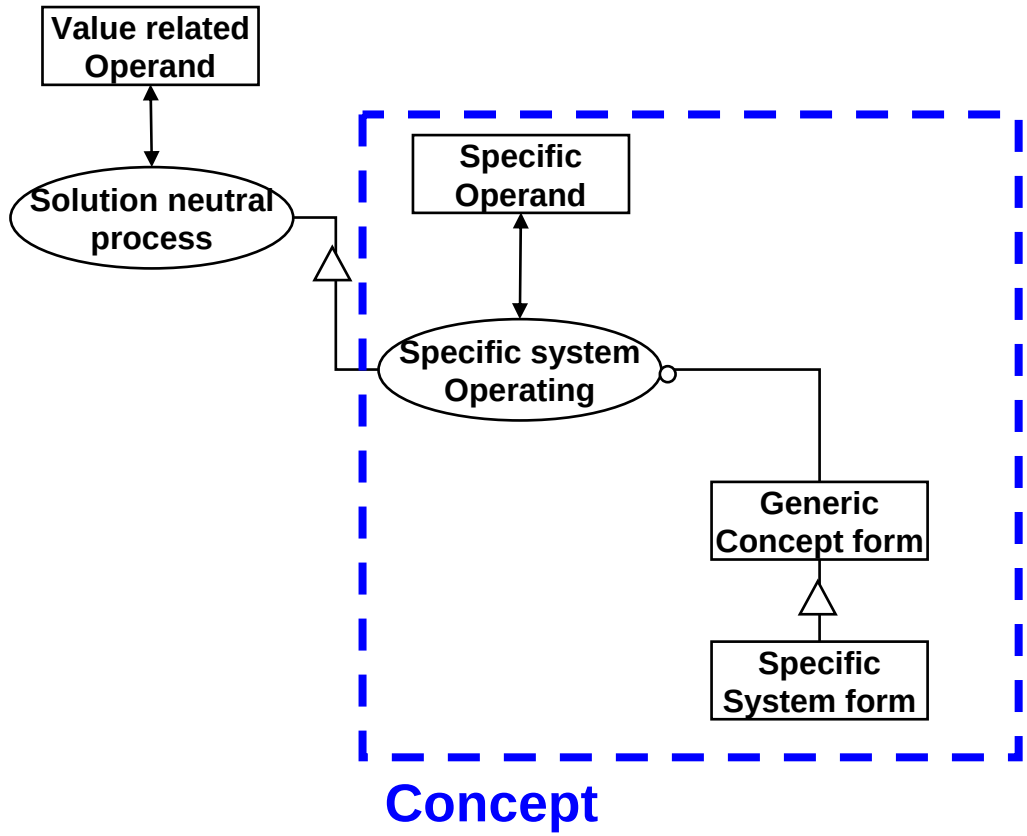
Concepts - Transporting

Solution neutral statement of function	Solution specific function	Solution specific abstraction of form
Transporting a person	Rolling	Car, wagon, skateboard
	Air flying	Aircraft, helicopter, glider
	Floating	Ship, surfboard, blimps, sailboat, canoe
	?	?

Concept Space - Questions

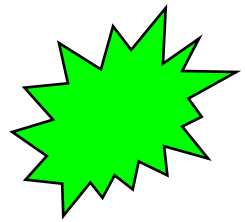


- **What is the specific operating process, and if necessary, the specific operand? (function concept)**
- **What is the generic instrument object that executes this process?**
- **What is the specialization of the instrument object? (form concept)**



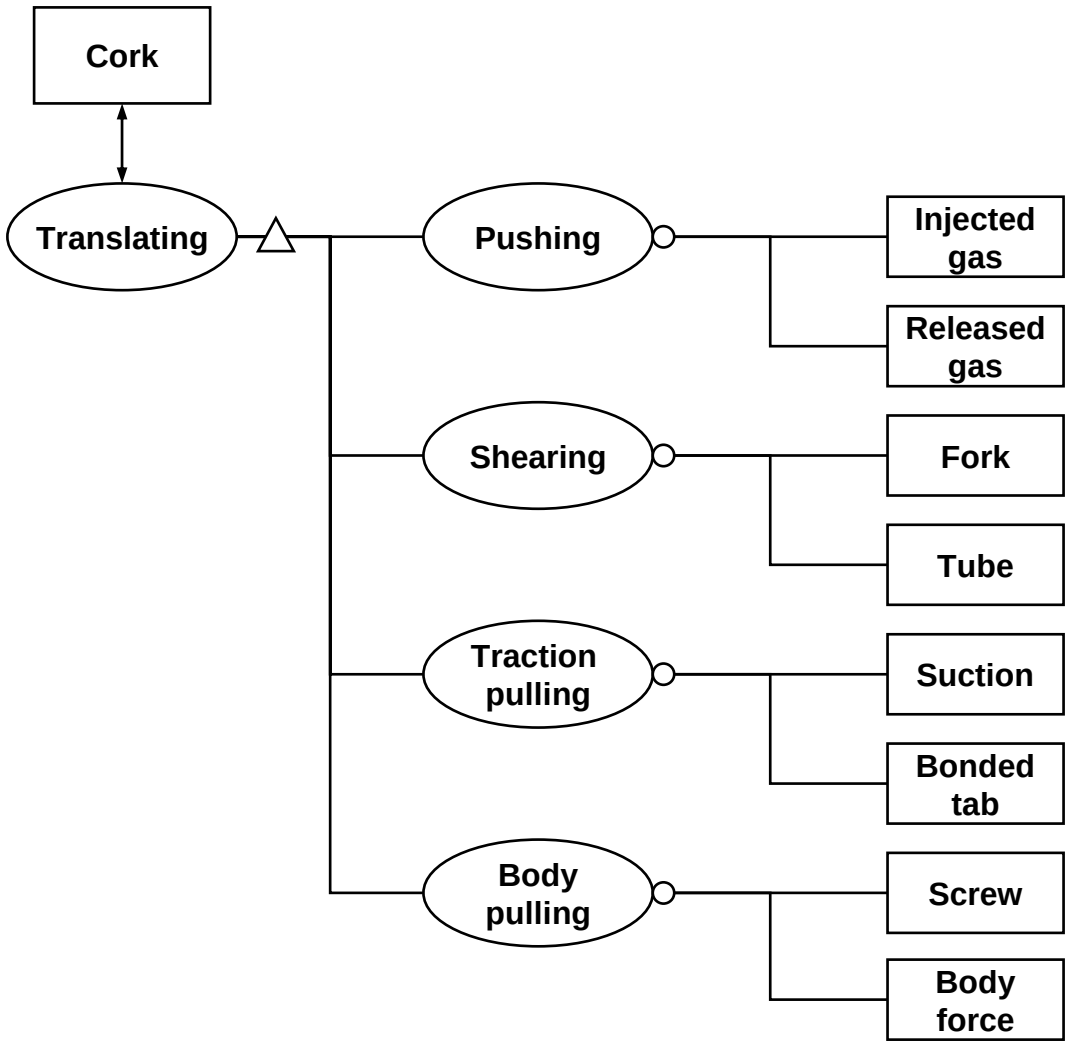
▲ Decomposes to
△ Specializes to
◭ Has attribute of

Exercise: Concepts for Fluid Extraction



- **Each group take an object and answer the concept questions:**
  - **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 the solution specific processes and operands that will achieve this transformation (process concept)?**
  - **What are the solution specific object that can act as instruments of this process (object concept)?**

Concepts for Fluid Extraction

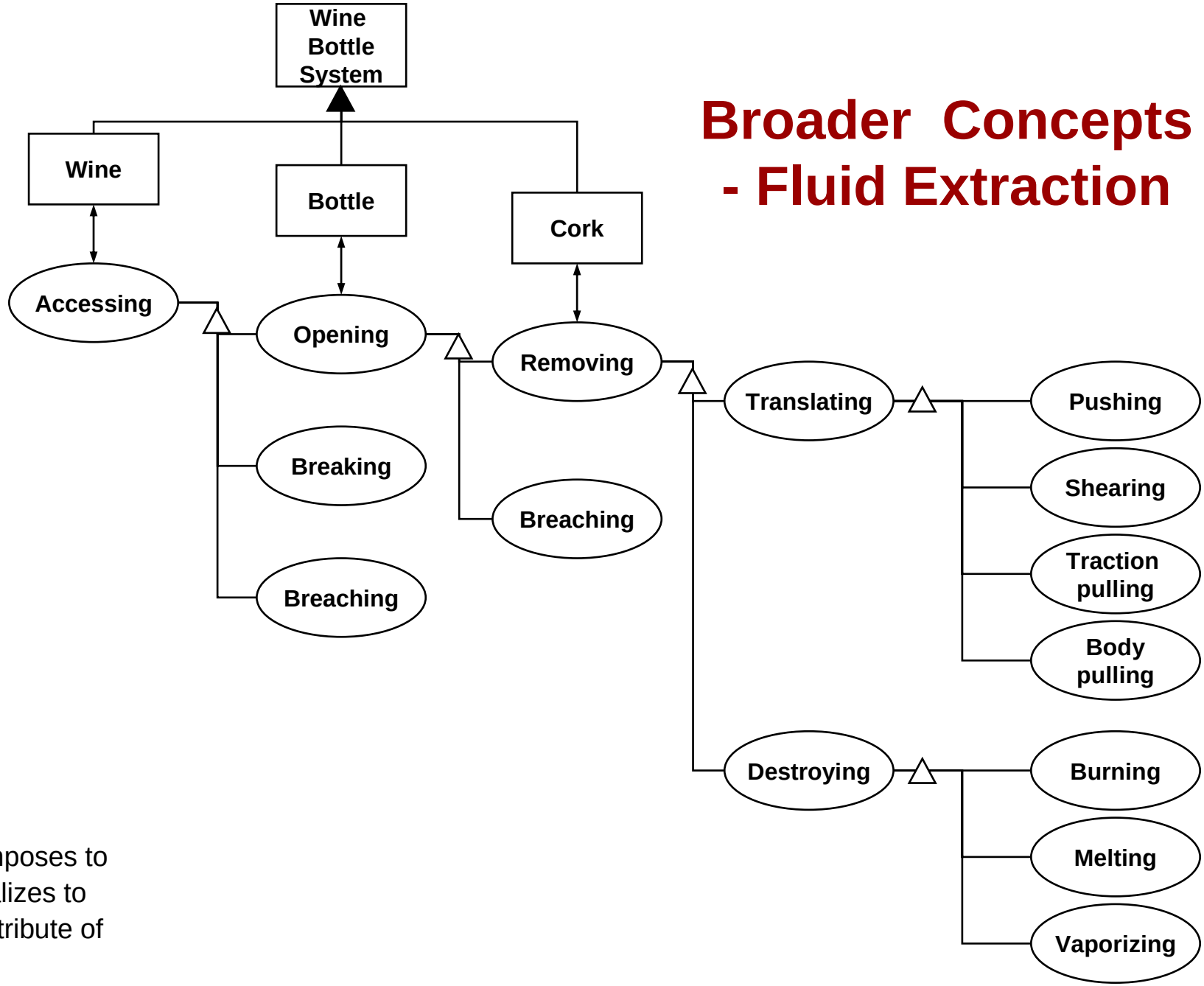


▲ Decomposes to
△ Specializes to
◬ Has attribute of

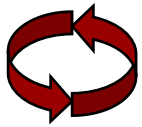
Broader Concepts and Recursion

- Often one can describe a specific operand, and the solution neutral transformation
- Or one can define the same problem at one or more higher levers of hierarchy recursively
- For example:
  - To increase shareholder value
  - To sell medical products
  - To sell medical sensors
  - To manufacture medical sensors
- It is often useful to represent this recursion in the concept tree

Broader Concepts - Fluid Extraction



Decomposes to
Specializes to
Has attribute of



Recursion

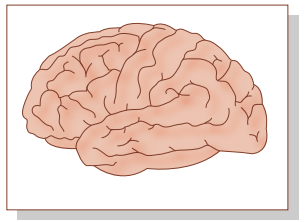
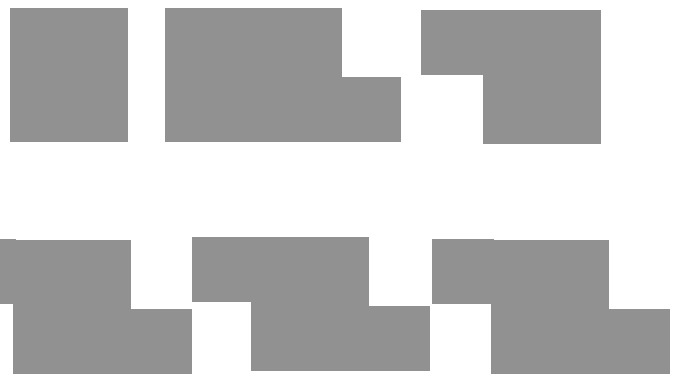
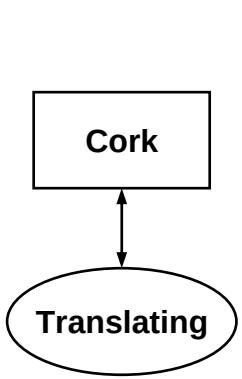


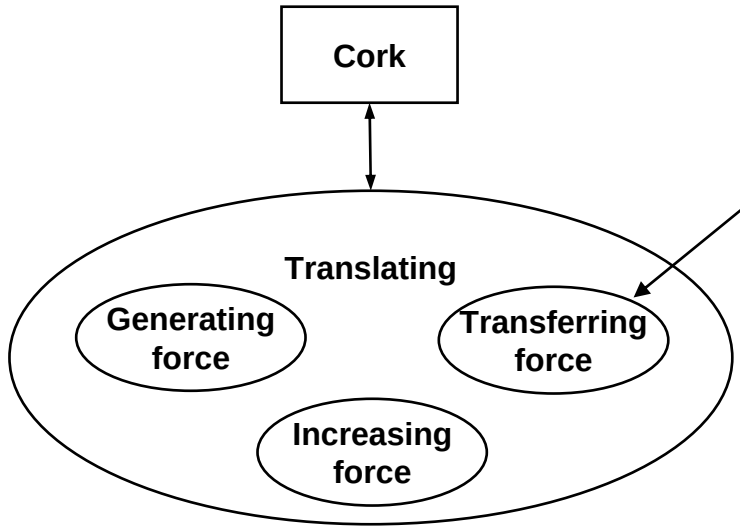
Figure by MIT OCW.

- **Recursion is the use of repeated steps or elements**
- **Processes can be used recursively**
  - **Turn left, turn right, turn left, turn right, …**
- **In a generalization, objects can be used recursively as well**
  - **No recursion**
  - **Recursion**



Multi-function Concepts for Fluid Extraction





Note: this is a interfacing process

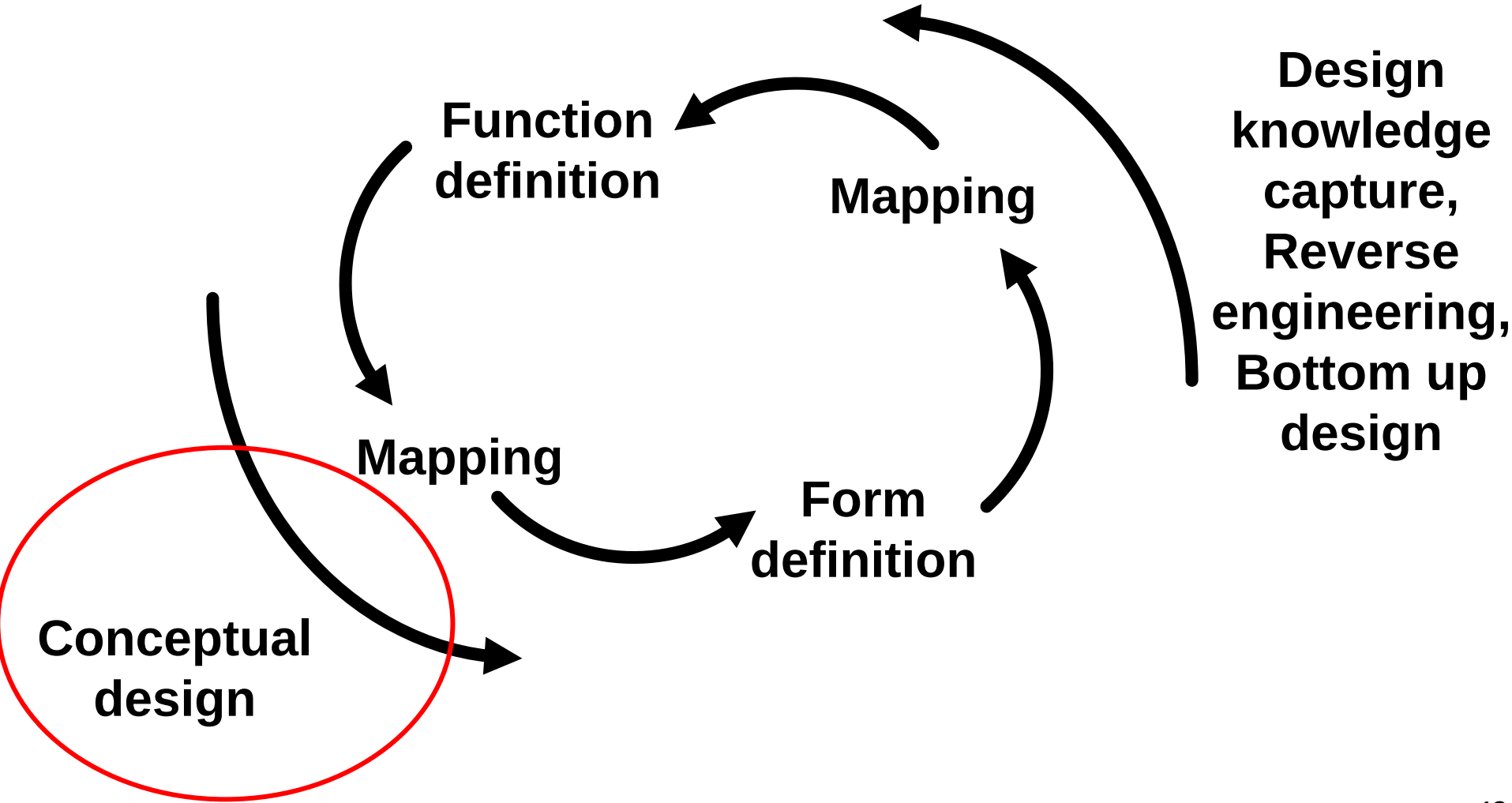
▲ Decomposes to
△ Specializes to
◭ Has attribute of

A process often, but not always, can be zoomed to reveal a set of internal functions, the emergence of which is synonymous with the process

Concept and Parameter Selection

- **When a concept is chosen, the *list* of design parameters is also implicitly established**
- **When the design parameters are chose, the design is finalized**
- **Products based on the same concept are continuously connected**
- **Products based on different concepts are disjoint**
- **Example: Table is concept - parameters are length, width, height, number of legs, etc. Counter is disjoint concept**

Form - Function Sequence



Sequences in Design and Reverse Engineering

- **In Design**
  - **Define externally delivered function, create concept, break down (zoom) internal function, define elements of form**

- **In Reverse Engineering**
  - **Define elements, infer concept, infer internal function, infer externally delivered function**
  - **Last two steps are difficult due to the emergence of function**

Expressing Concept

- There is no convention or standard for naming concepts, but they tend to be nouns or noun phrases
- Rationally, they should be named “operand + process + instrument” but few are
- They are often named by “operand + process + er” (lawn mower), but this often works only for the first such device (e.g. what is a people mover?)
- Other common patterns are the operand + instrument, or even just instrument
- New concepts can be expressed by a few words or a short phrase (e.g. cell phone)
- Established concepts can often be expressed by a word or two or an icon (e.g. Refrigerator, )

	Operand	Process	Instrument
Operand/Process	lawn	mow	er [rotary]
	hair	dry	er [portable electric]
	phone	tele	[cordless]
Operand/instrument	light	(producing)	bulb
	cork	(removing)	screw
	fire	(burning)	place
	hat	(storing)	rack
	suit	(carrying)	case
Process/instrument	(data and info)	compute	er
	(article)	carrying	case
Process	(TV)	control [remote]	(device)
	(painting)	painting	(paint)
Instrument	(head)	(covering)	hat
	(food)	(serving)	table
	(person)	(carrying)	bicycle

Summary of Concept

- A system vision which maps form to function
- It involves a principle of operation and an abstraction of form
- It rationalizes the details of the architectural structure
- Is created by the architect
- Must allow for the execution of all functions
- Specifies the vector of design parameters, which, when selected, will establish the design

Creativity

- **Defined: The ability or power to cause to exist, to bring into being, to originate, or to combine in a new way**
- **Focusing creativity is a role of the architect**
- **Innovative new architectures often build around a creative new idea**
- **The concept development process is often a time of peak creativity**
- **Creativity must be tempered by the need to get something accomplished**

Types of Creativity

- **Raw or pure creativity - thinking of something that no one has ever thought of - This is *rare***
- **Transfer of experience or metaphor from one field to another - Very common**
- **Organizing knowledge, finding patterns, interpolating and extrapolating - Even more common**

Approaches to Stimulating Creativity

- **Study previous work (reverse engineering, benchmarking, patent search, etc.)**
- **Metaphors from other systems (e.g. nature)**
- **Group Dynamics (brainstorming, six hats)**
- **Structural processes (TRIZ, mind mapping)**
- **Intellectual stimulants (provocation, motion)**

Summary Creativity

- **We learned there are about four ways to stimulate creativity in concept design:**
  - **Metaphor (physical or human-made)**
  - **Invention (or new science)**
  - **Combination, rearrangement, evolution**
  - **Patterns and pattern matching**
- **Objective is to move off established neural pathways!**

References on Creativity

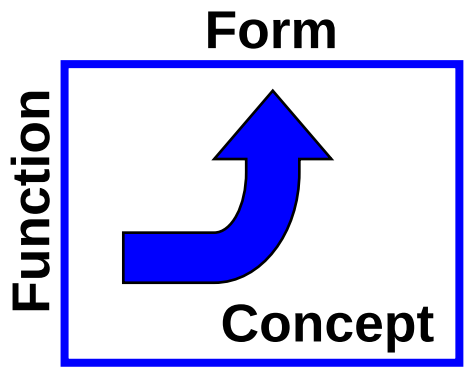
- **TRIZ web sites (e.g. www.jps.net/triz/triz.html)**
- **Edward deBono: Lateral thinking, Serious Creativity**
- **Notes on the Synthesis of the Form, Christopher Alexander, Harvard University Press, 1964**
- **Integrated Methods for Successful Product Engineering, Pugh**

Today's Topics

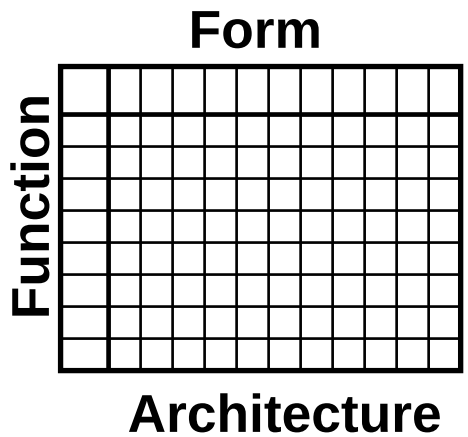
- **Reflection on Function**
- **Concept**
- **Creativity**
- **Architecture**
  - **Vs. concept**
  - **Analysis through to internal value related processes**
  - **Inference from form**
  - **How do they connect to produce architecture?**
- **Etc.**

Concept vs. Architecture

- **Concept is a project or system vision, idea, notion or mental image which includes the principle of operation and abstraction of form, and therefore maps Function to Form**



- **Architecture is the details of the assignment of function to form, and the definition of interfaces and structure**

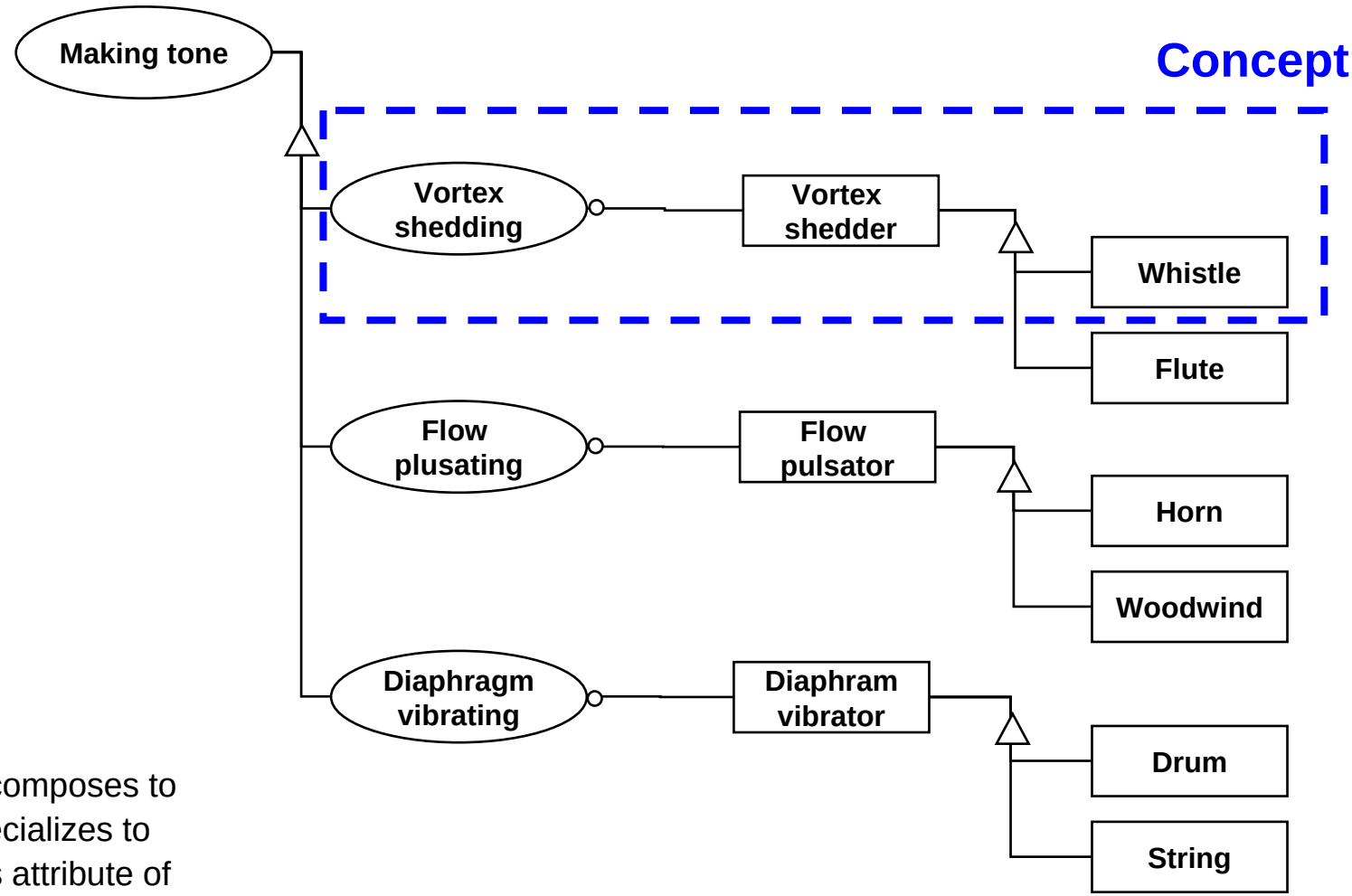


- **We still lack a notation to give this any coherence**

Begin Architecture with Value Related Process "Flow" Analysis

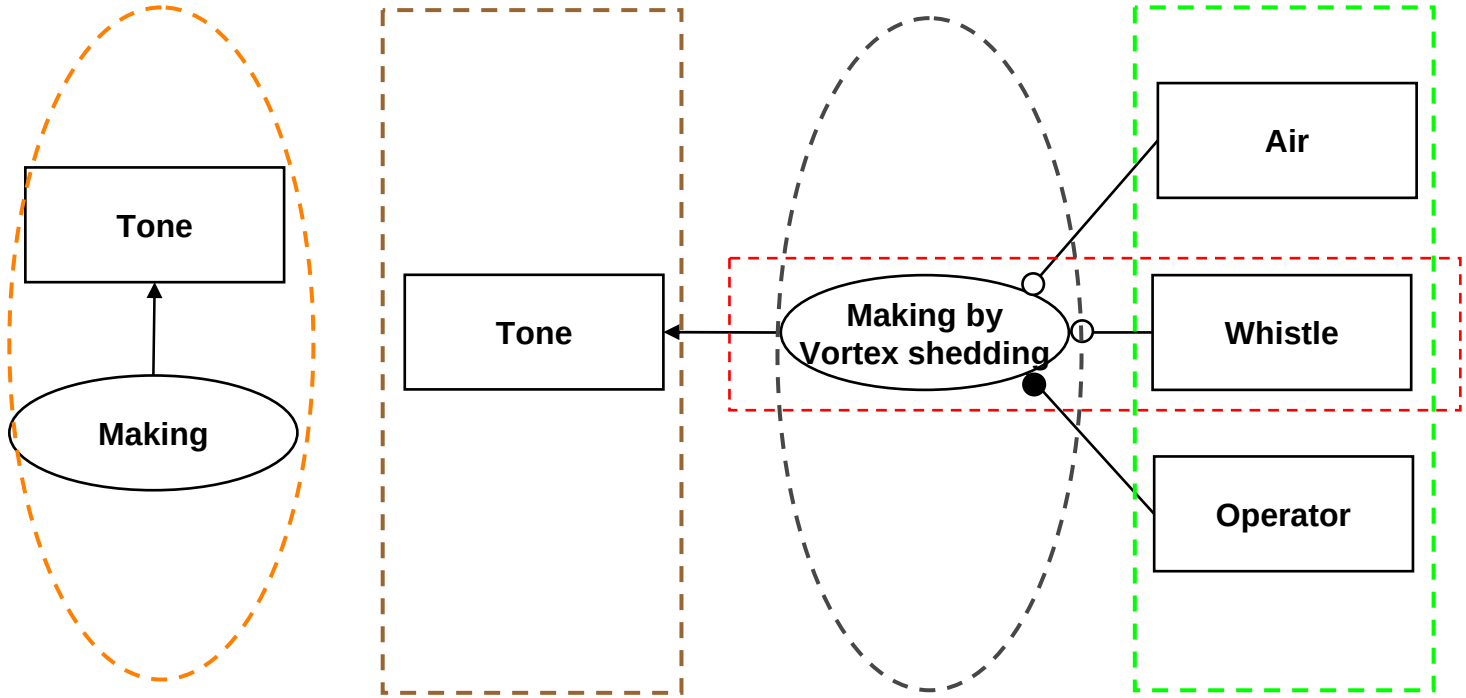
- **Start the analysis of architecture by identifying the solution neutral statement of function, concept (and potential multifunctional aspects) and whole product system**
- **Then immediately begin by identifying the main "flow" of processes and operands within the product system that creates value - the internal value related processes**
- **Only then, try to connect the objects of form and their structure to the internal processes that deliver value**

Concepts - Making Tone



▲ Decomposes to
△ Specializes to
◭ Has attribute of

Tone Making Concept - Whistle



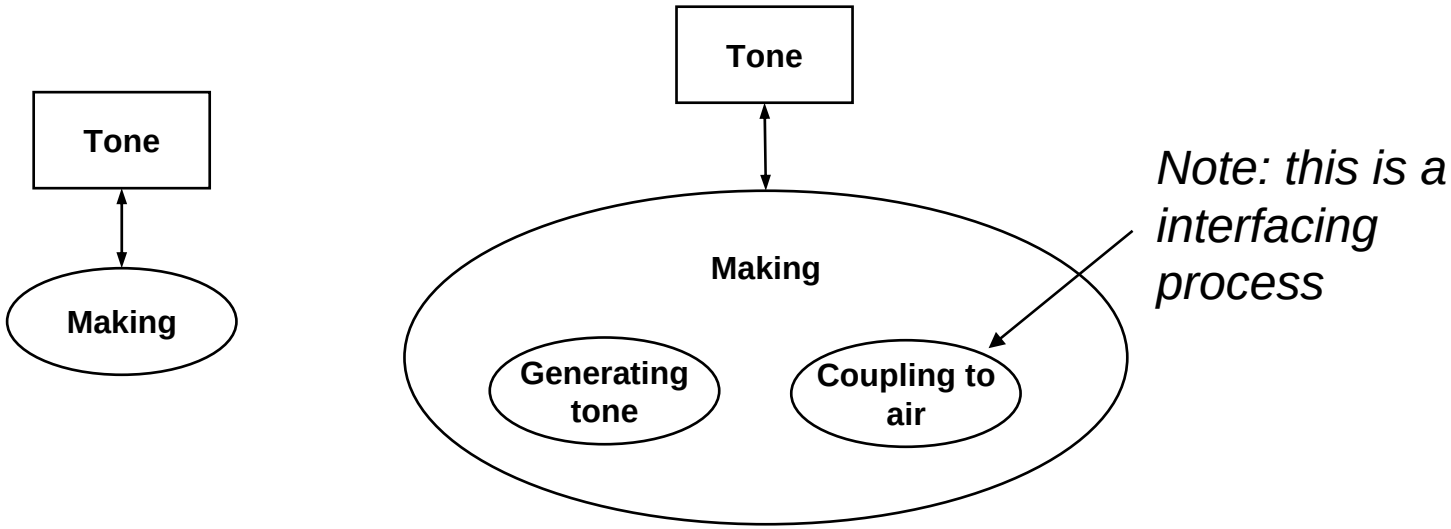
Operand and solution neutral transformation

Specific operand - value related external object that changes

Externally delivered value related process

Value related instrument objects

Multi-function Concepts for Making Tone



Making a tone more or less immediately breaks down into generating the tone, and coupling it to the air

▲ Decomposes to
△ Specializes to
◭ Has attribute of

Whistle - Idealized internal value related processes

- Idealized internal value related processes and operands informed by the concept whistle

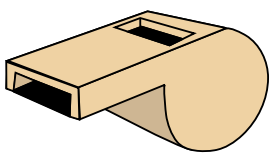
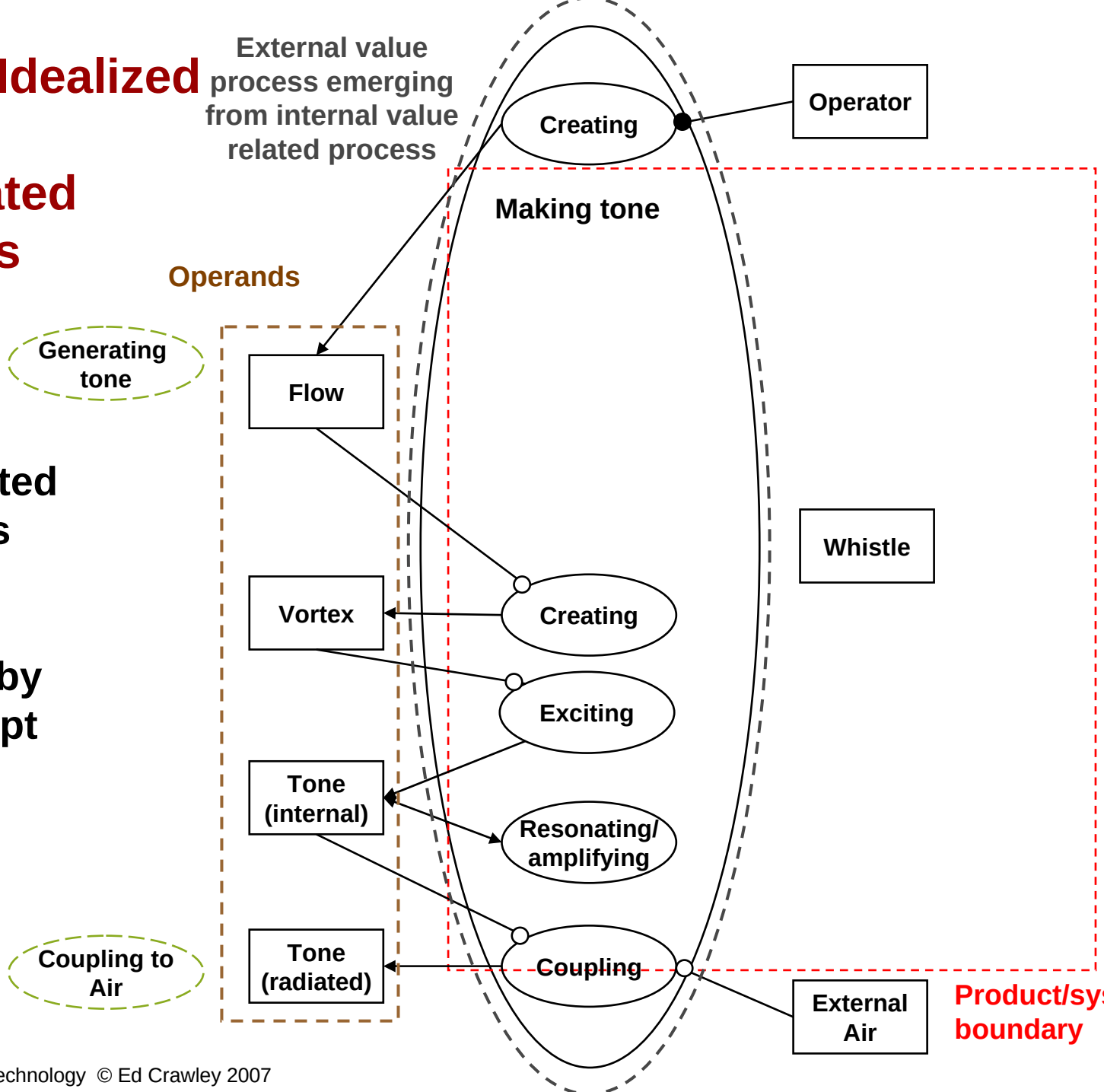


Figure by MIT OCW.



Whistle - Realizable Internal value related processes

- More realizable internal value related processes and operands informed by the concept whistle

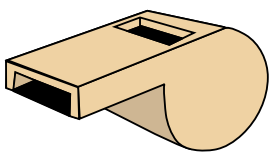
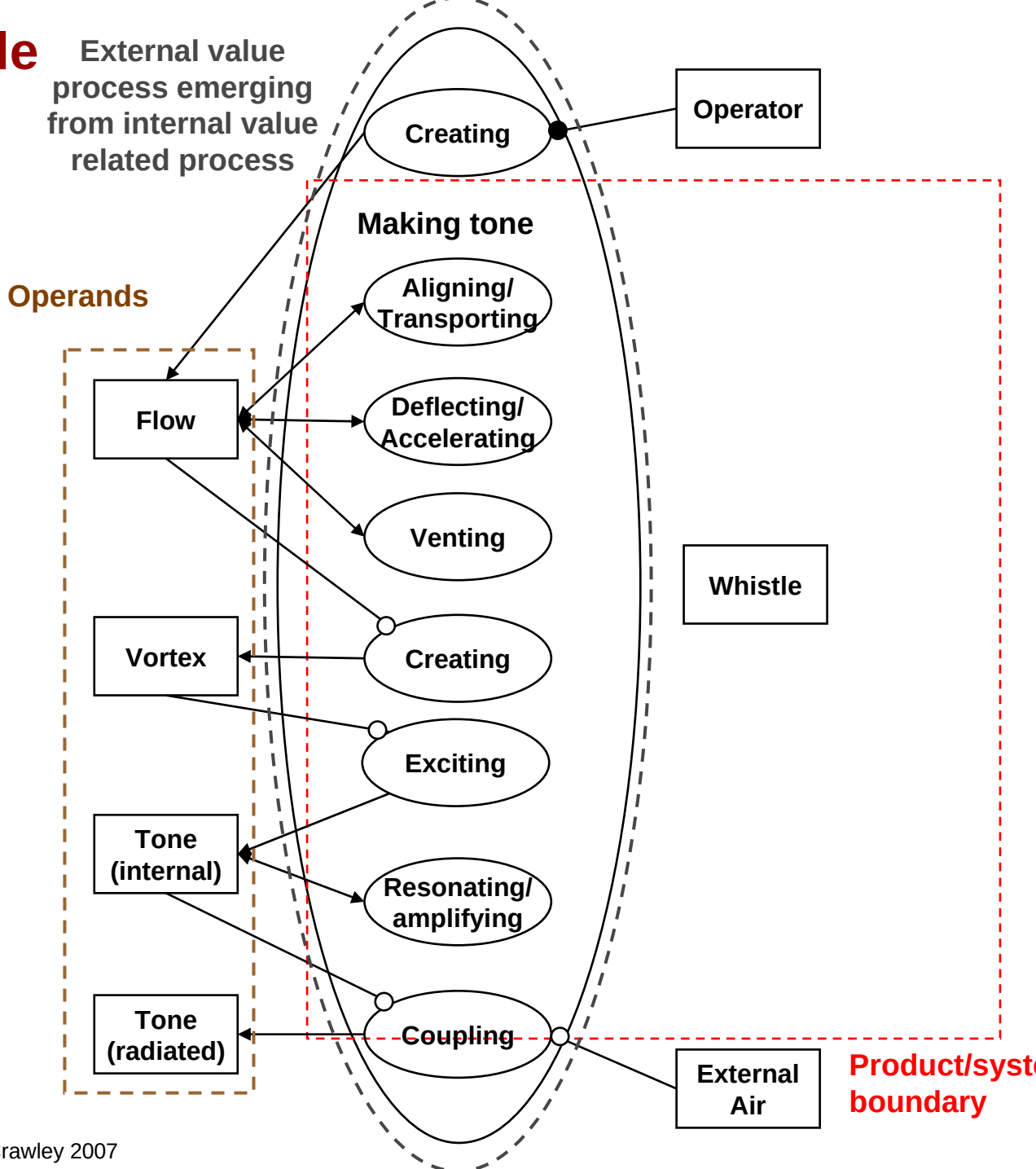
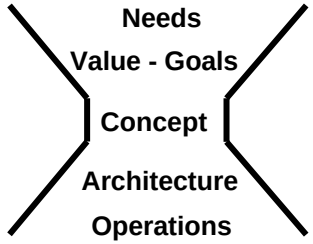


Figure by MIT OCW.

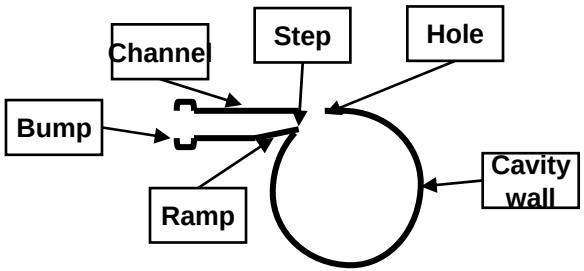


Expanding a Concept to an Architecture (1)



- Identify the operand, and value related attribute, and solution neutral transformation
- Identify the concept process and instrument object, and other aspects of the whole product system and use context
- Identify aspects of multifunctional concepts, if applicable
- Informed by the concept form, identify the:
  - Idealized internal processes that touch directly on the delivery of value - the “value related internal processes”
  - The intermediate operands along that path
  - (perhaps) More realizable internal value related processes
- Begin to make an OPM of the architecture

Process - Object Architecture - Whistle



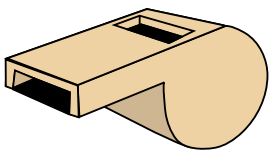


Figure by MIT OCW.

External value process emerging from internal value related process

Creating

Operator

Value related instrument object

Making tone

Whistle

Operands

Aligning/ Transporting

Bump

Flow

Deflecting/ Accelerating

Channel

Ramp

Venting

Step

Vortex

Creating

Hole

Exciting

Cavity wall

Tone (internal)

Resonating/ amplifying

Star

Ring

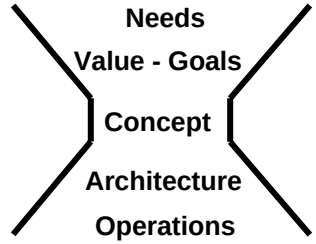
Tone (radiated)

Coupling

External Air

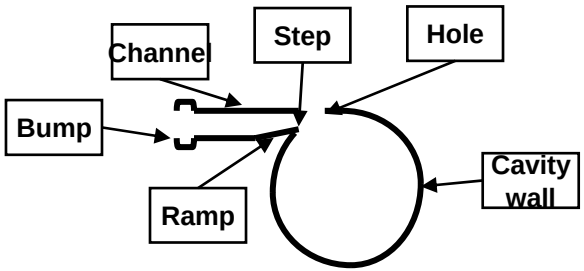
Product/system boundary

Expanding a Concept to an Architecture (2)



- Now,
- Within that concept, identify the:
  - Instrument objects to execute the internal value related processes
  - The other instruments that are necessary to deliver value - the whole project system
- Continue to make an OPM of the architecture

Whistle: Process - Object Architecture



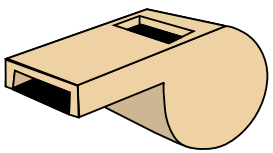
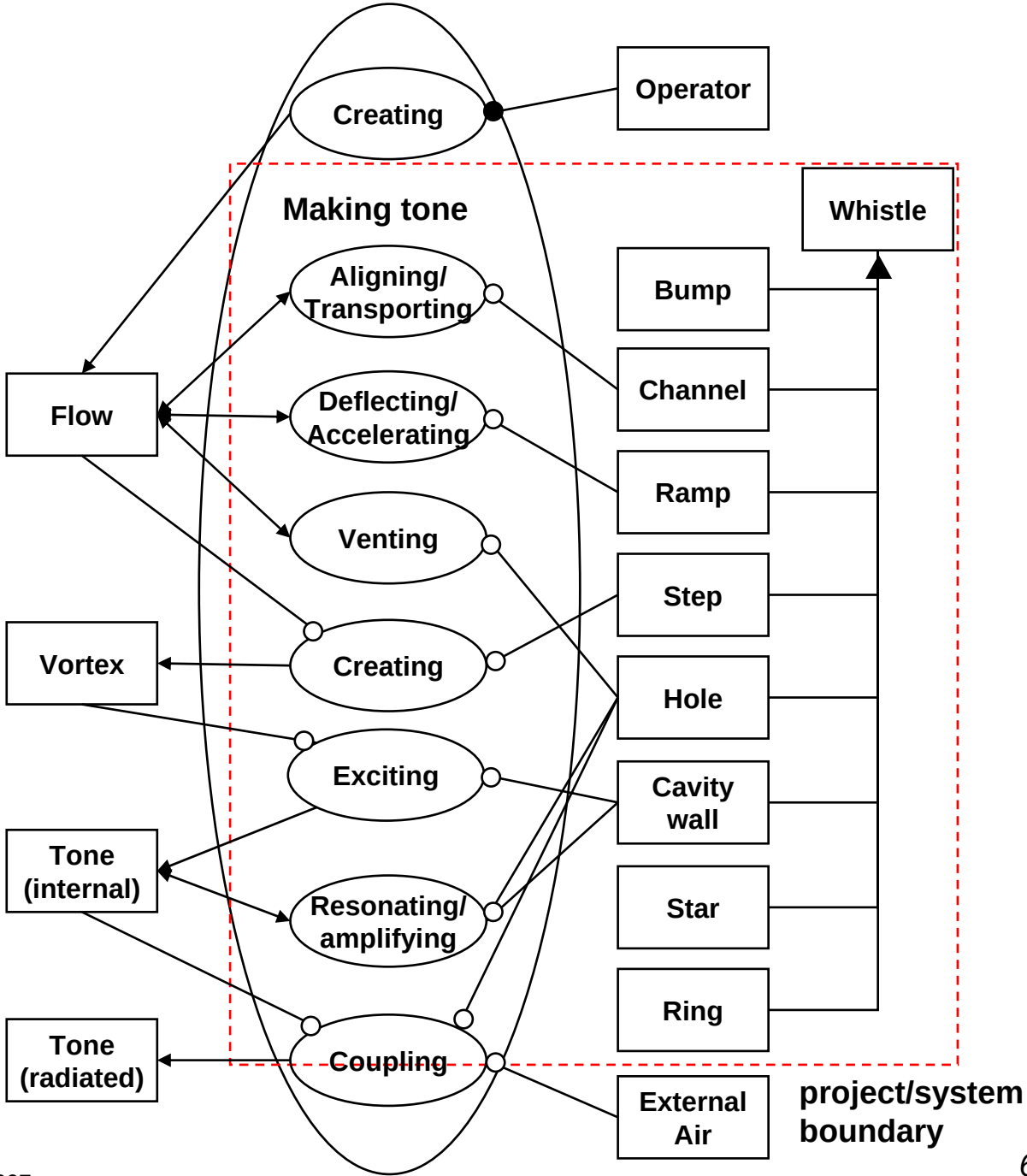
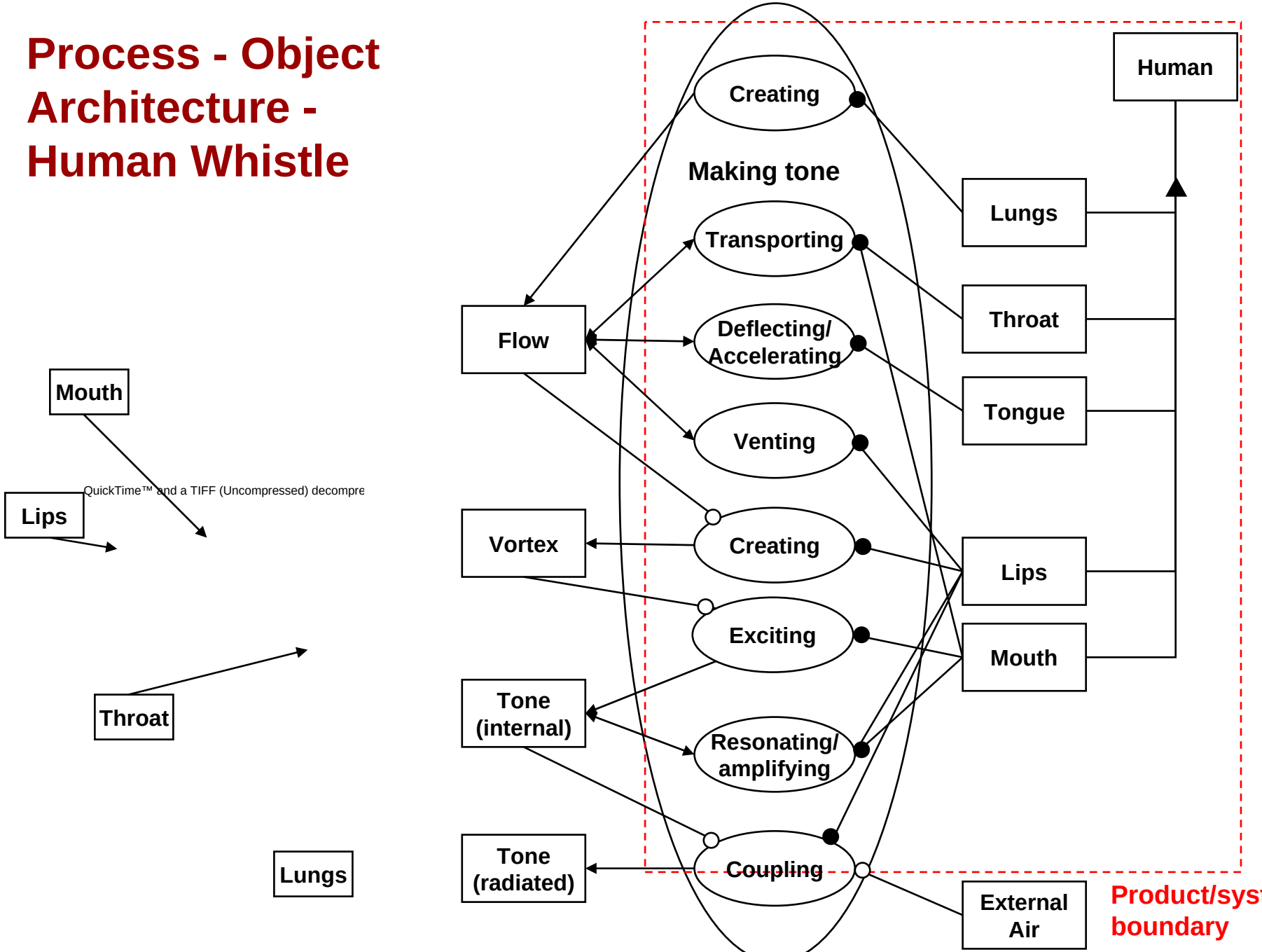


Figure by MIT OCW.



Process - Object Architecture - Human Whistle



Observations:

- **The selected concept has been analyzed for multifunctional aspects**
- **Informed by the concept form, the internal idealized and realizable value related processes have been developed**
- **A more detailed decomposition of form has been identified, which mapped onto the function**

- **Two remarkable different forms are found to have the same internal processes, but different mappings, which makes them different architectures!**

Process-Object Architecture Matrix

	Instrument Object 1	Instrument Object 2	Instrument Object 2	Operand 1	Operand 2
Process 1	relationship	relationship		relationship	
Process 2		relationship	relationship	relationship	relationship
Process 3			relationship		relationship

- **Graph of Process - Object Architecture can be converted to a matrix**
- **Objects - Instruments *and* Operands on one side**
- **Processes on the other**
- **Relationship indicated by text - Full NNV structure**
- **Not symmetric, but causal (cause and effect implied)**

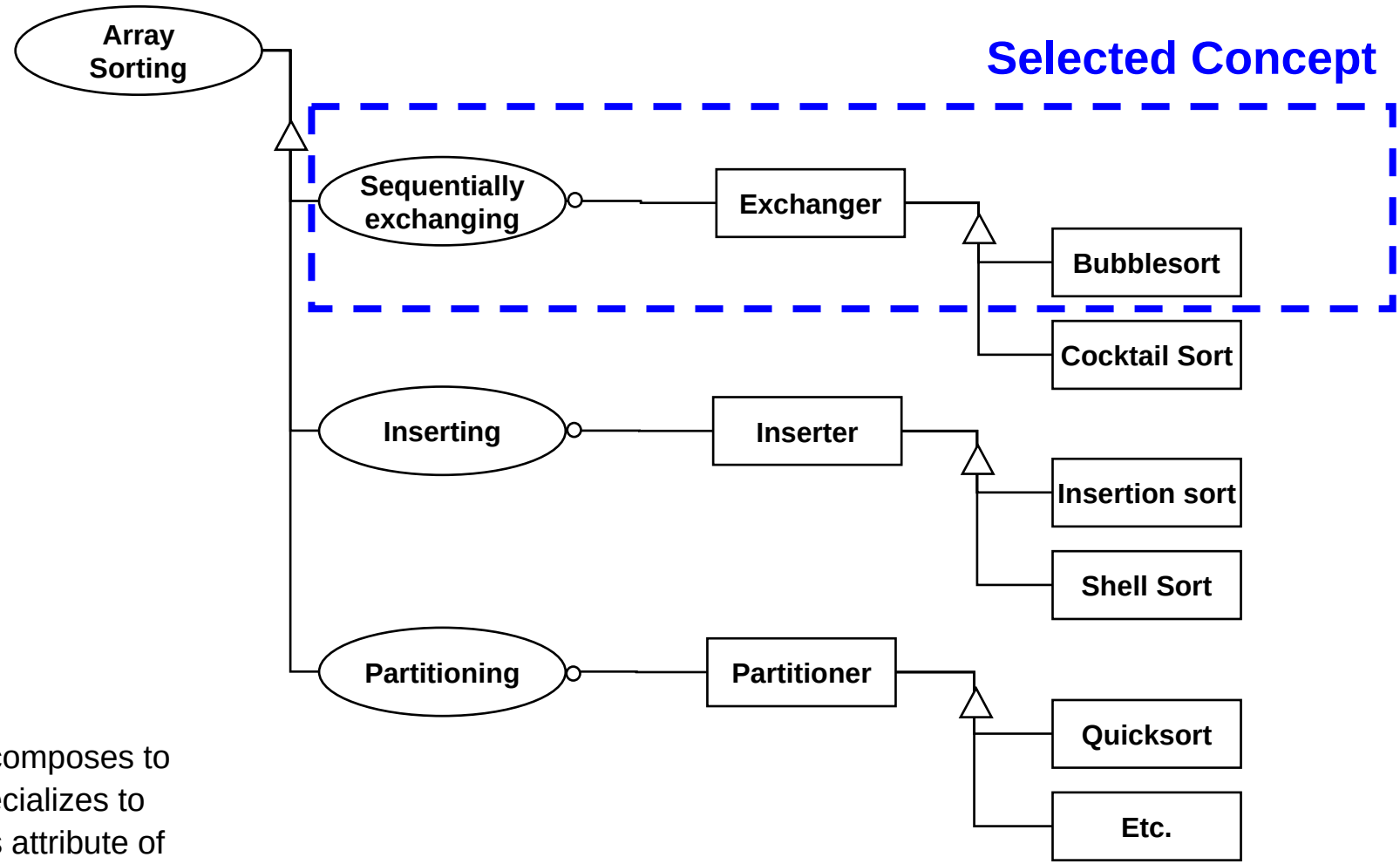
Process-Object Architectures - Whistle

Plastic Whistle	operator	bump	channel	ramp	step	hole	cavity	star	air	flow	vortex	tone (in)	tone (ex)
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

Human Whistle	lungs	throat	tongue	lips	mouth	air	flow	vortex	tone (in)	tone (ex)
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 = agent, I = instrument, c = creates, e = effects

Concepts - Sorting Array

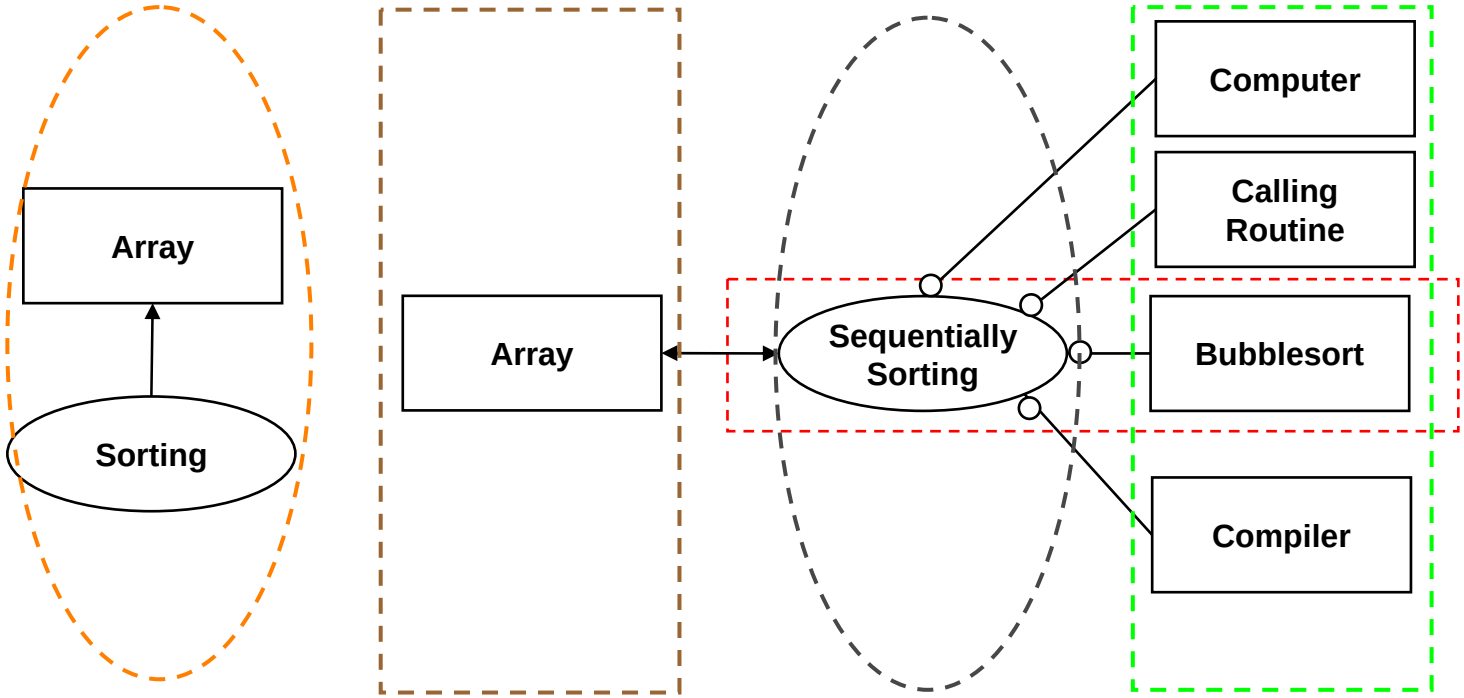


Product/System - Code Bubblesort

```
Procedure bubblesort (List array, number length_of_array)
    for i=1 to length_of_array
        for j=1 to length_of_array - i
            if array[ j ] > array [ j+1 ] then
                temporary = array [ j+1 ]
                array[ j+1 ] = array [ j ]
                array[ j ] = temporary
            end if
        end of j loop
    end of i loop
    return array
End of procedure
```

Product/system boundary

Sorting Array Concept - Bubblesort



Operand and solution neutral transformation

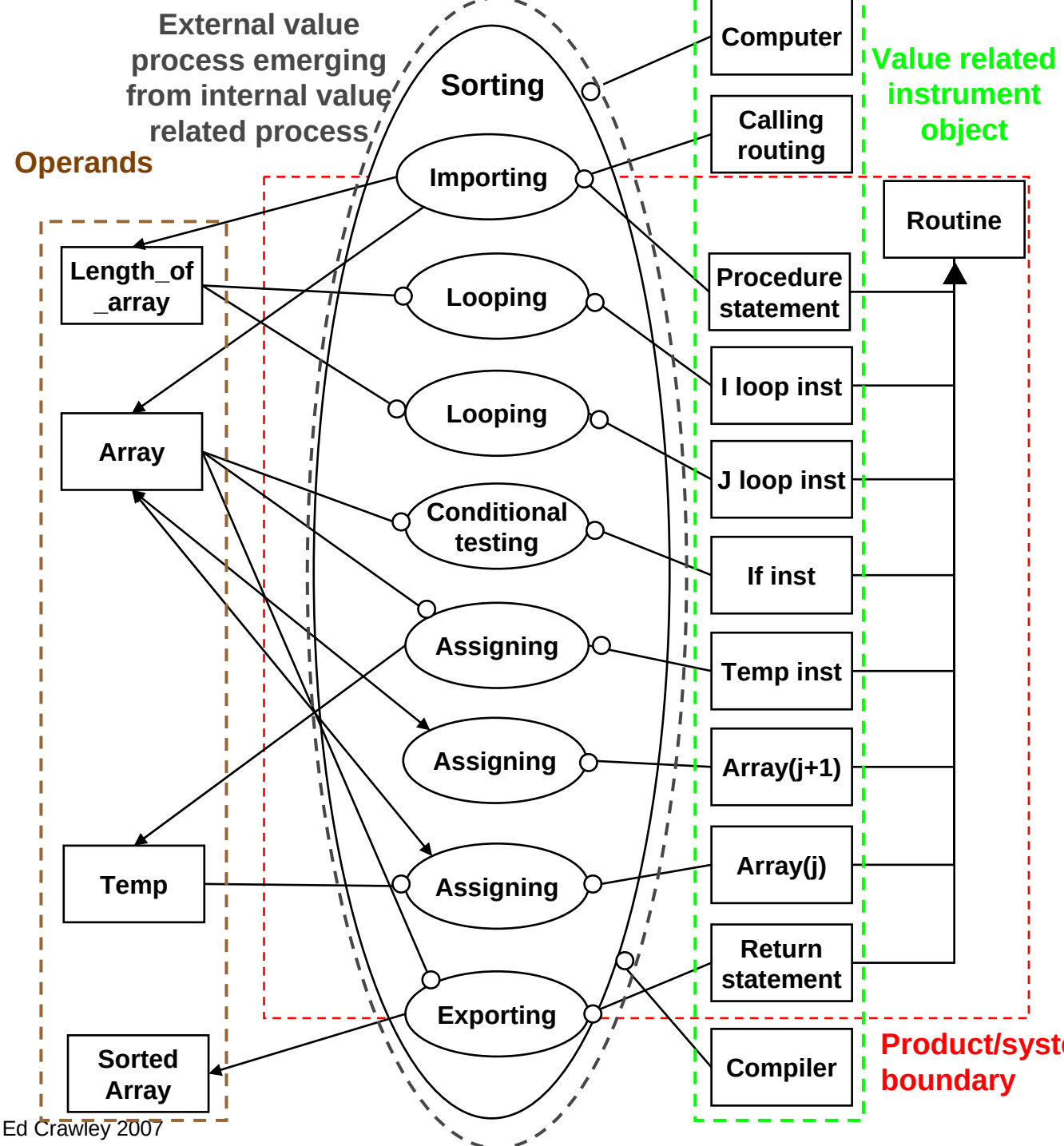
Specific operand - value related external object that changes

Externally delivered value related process

Value related instrument objects

Software Code Bubblesort : Process - Object Architecture

- Note: process control is missing
- How would you show this?



Reflections on Simple System and Scale Up to Medium Complexity Systems

- **Up to now, we have examined simple systems (whistle, op amp, bubblesort, corkscrew, plus OS 2) largely for pedagogic reasons, to understand ideas unencumbered by complexity**
- **Now we will start to examine "medium" complexity systems (skateboard, refrigerator, ServeCo, TCP, plus OS 4), to start to develop the means to examine complex systems**
- **One challenge is that the value flow analysis and the analysis of the form elements and structure do not immediately or obviously connect, have to work harder at matching these up by using "*outer in*" thinking**

Types of "Vertical" Thinking

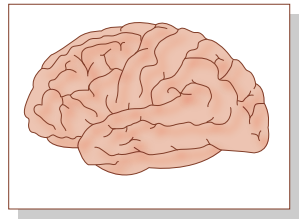
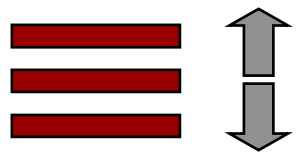


Figure by MIT OCW.

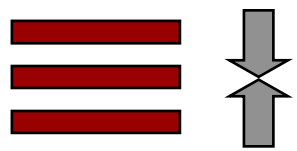
- **When examining complex systems, there are several ways of thinking through them:**
- ***Top down* - start at the "highest" level and reason down through the system**

- **Bottom up - start at a lower level and reason up**

- **Middle out - start at a middle level, and reason toward the top and bottom**



- **Outer in - start at a the top and bottom, and reason toward the middle**



Example - Skateboard

- **Simple mechanical product/system**
- **Truly only a medium complexity system (about 20 part types and about 70 total parts)**
- **Model for *transporting* as primary externally delivered function**

Concept to Architecture - Skateboard

- Identify the operand, and value related attribute, and solution neutral transformation
- Identify the concept process and instrument object, and other aspects of the whole product system and use context
- Identify aspects of multifunctional concepts, if applicable
- Informed by the concept form, identify the:
  - Idealized internal processes that touch directly on the delivery of value - the “value related internal processes”
  - The intermediate operands along that path
  - (perhaps) More realizable internal value related processes
- Within that concept, identify the:
  - Instrument objects to execute the internal value related processes
  - The other instruments that are necessary to deliver value - the whole project system

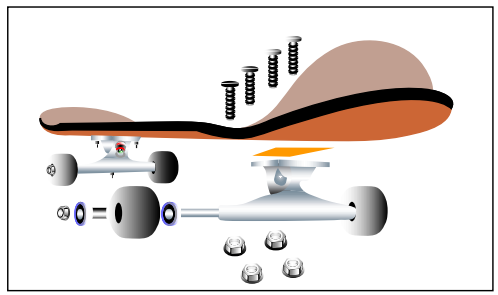
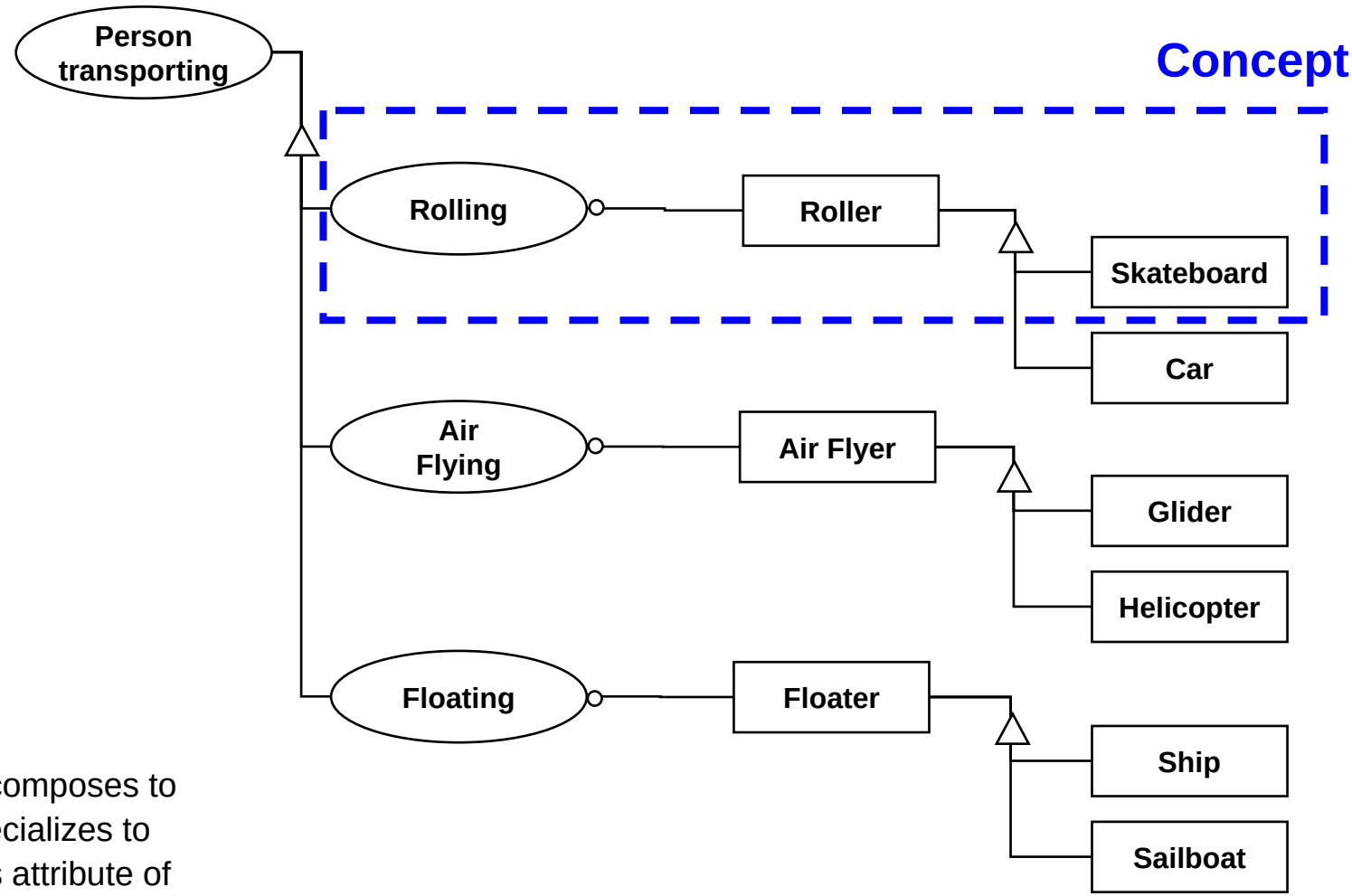
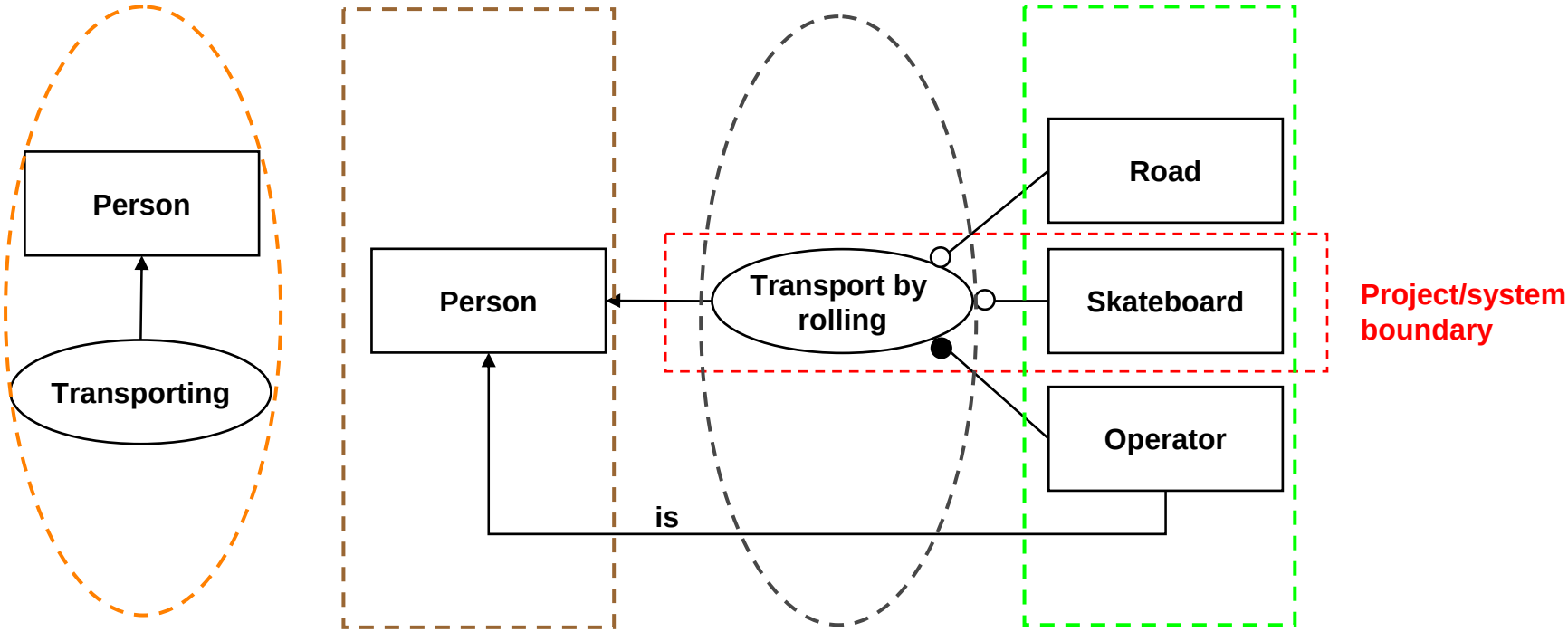


Figure by MIT OCW.

Concepts - Transporting a Person



Transporting Person Concept - Skateboard



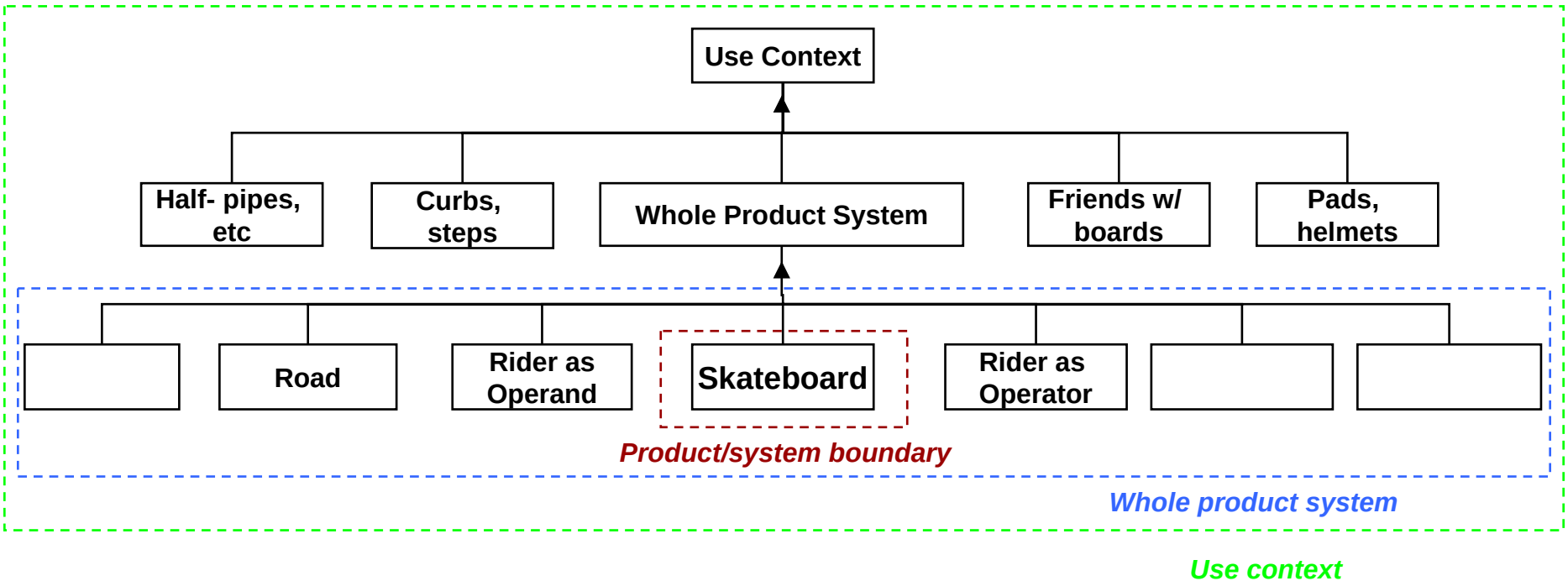
Operand and solution neutral transformation

Specific operand - value related external object that changes

Externally delivered value related process

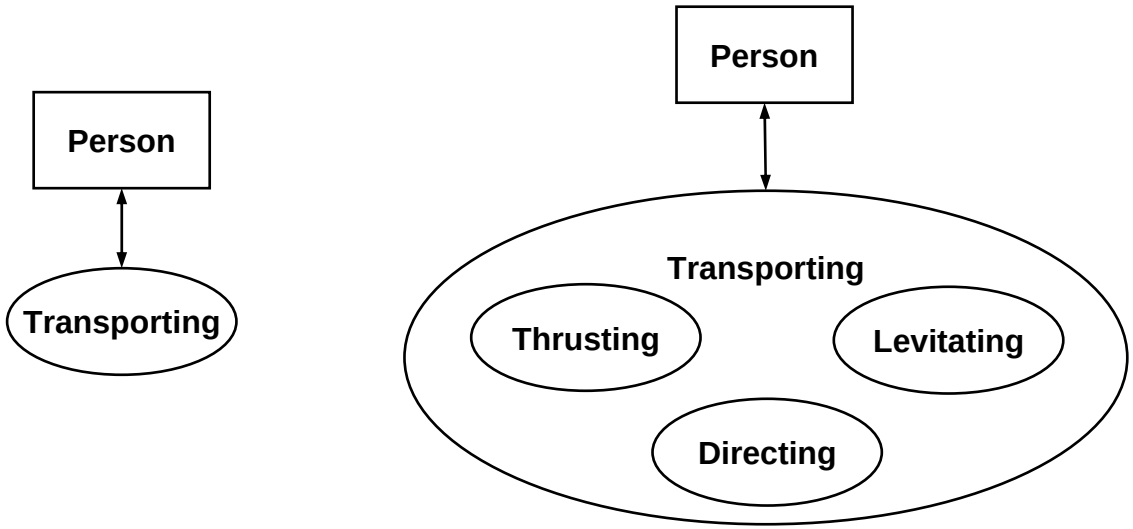
Value related instrument objects

Use Context - Skateboard



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

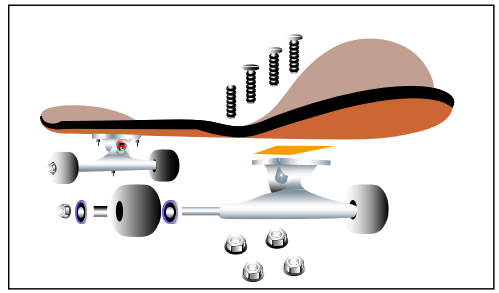
Multi-function Concepts for Transporting



▲ Decomposes to
△ Specializes to
◭ Has attribute of

Transporting contains three important sub-functions: overcoming drag (thrusting), overcoming gravity (levitating) and controlling the path of motion (directing)

Figure by MIT OCW.



Operand

is

Person

Pushing

Rider

Primary value process

Thrusting

Transporting

Levitating

Directing

Rolling

Steering

Skateboard

Road

Product/system boundary

Skateboard - Idealized internal value related processes

Internal value related process

Skateboard

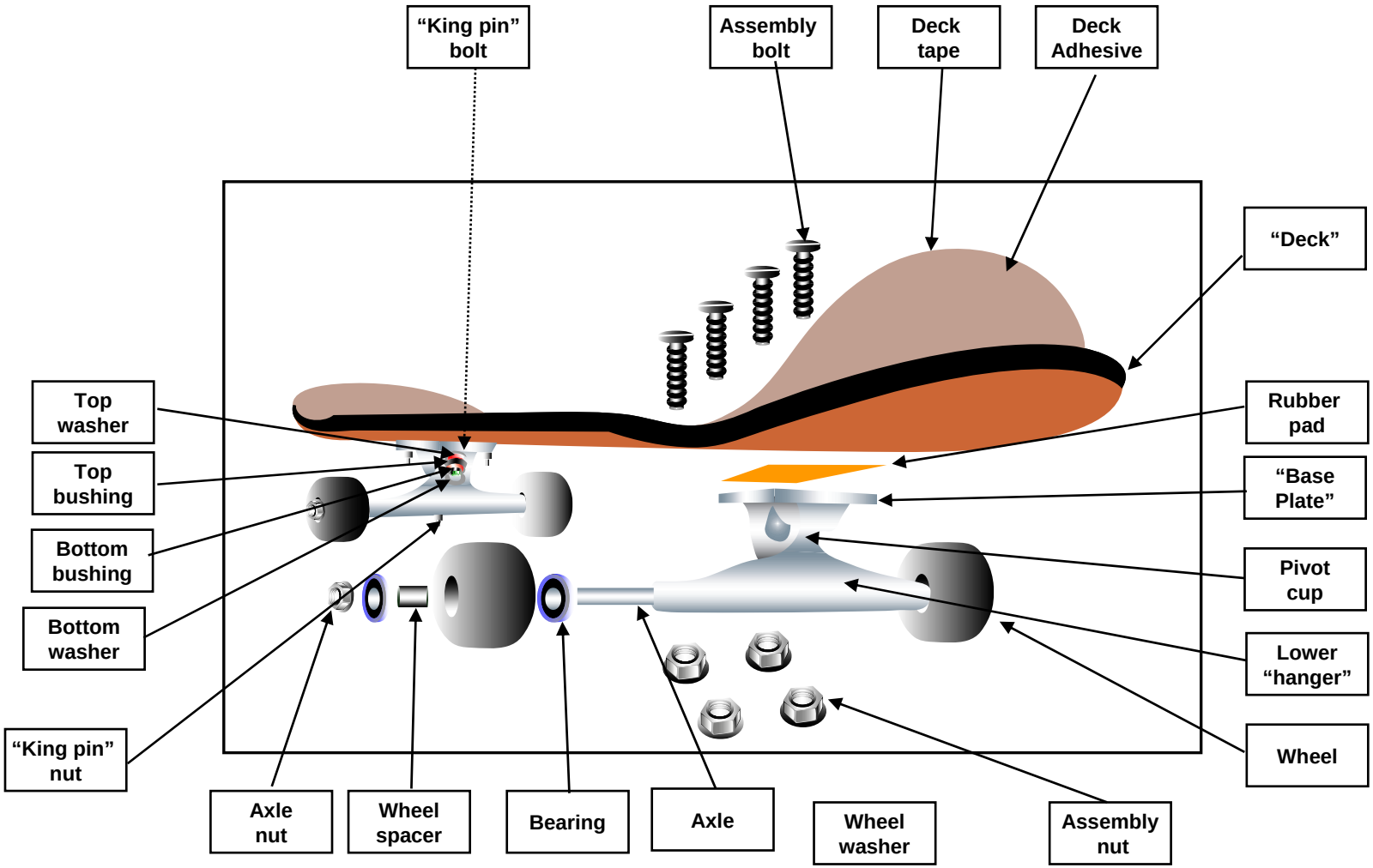


Figure by MIT OCW.

Decompositional View of a Medium System - Skateboard

- **Skateboard is composed of about 69 elements of 21 types**

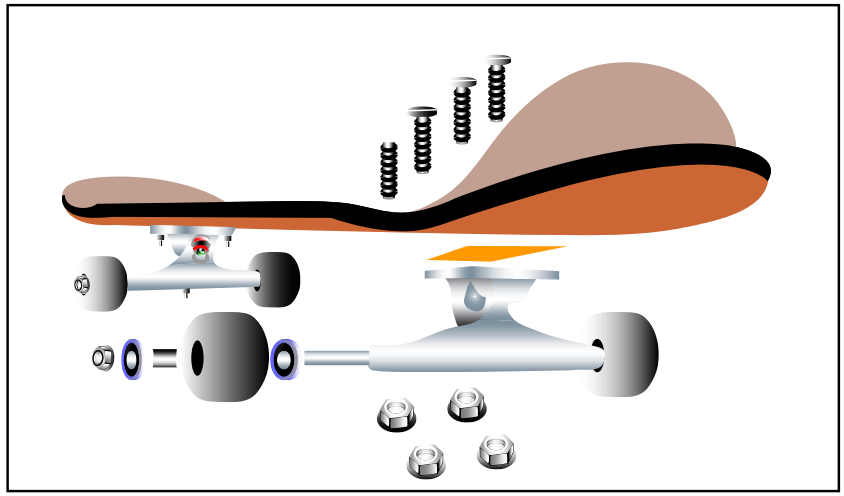
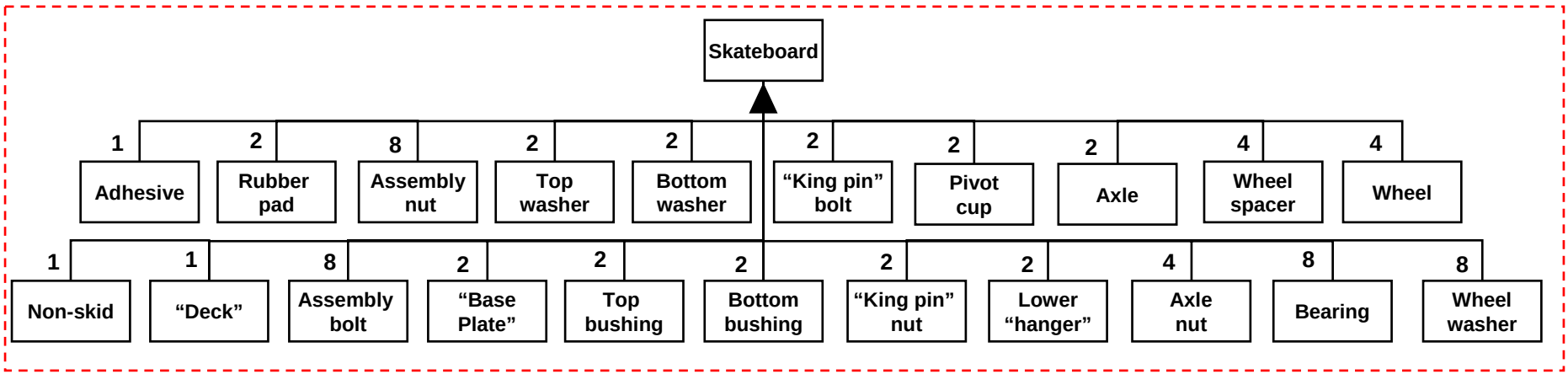


Figure by MIT OCW.

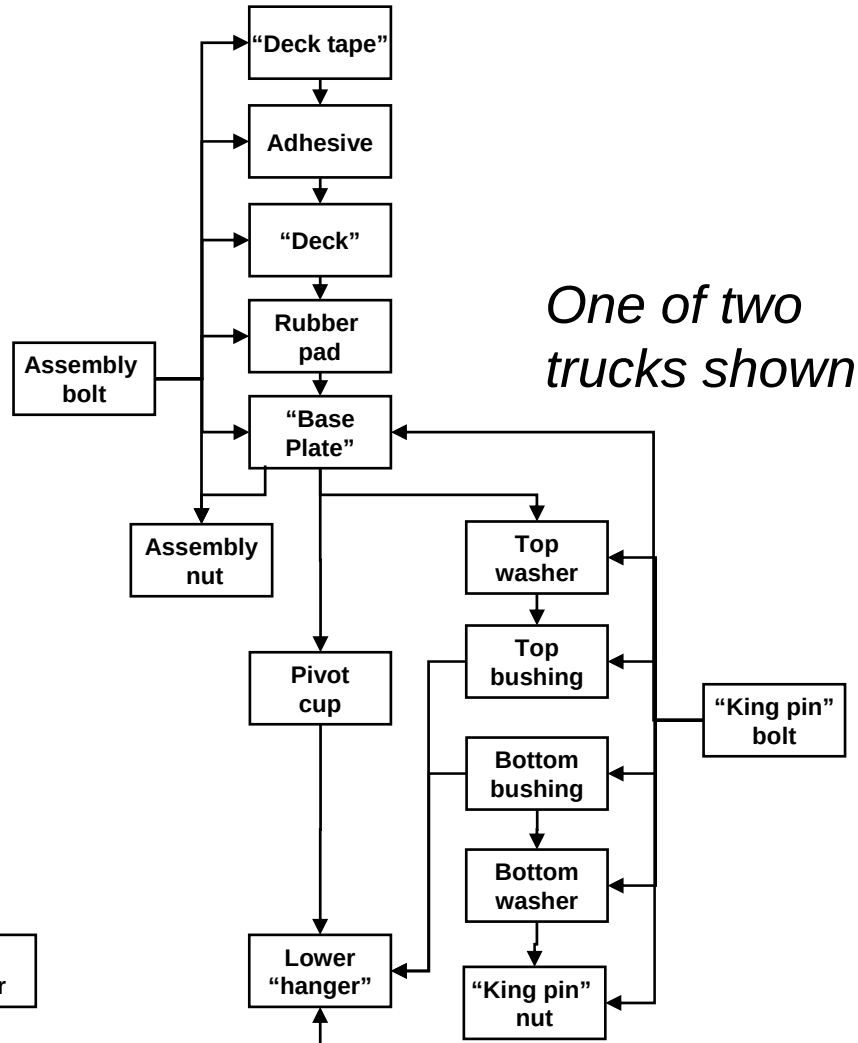
Product/system boundary



Each bearing could be decomposed to an inner and outer race, balls (8) and ball retainers (2) for 157 elements of 24 types

Spatial/Topological Structure - Skateboard

- **For physical systems, this information could also be shown on an assembly diagram, but topology would not be as explicit**
- **In a complete description, these arrows would be labeled**



One of two trucks shown

One of four wheels shown

Spatial Structure - “List” - Skateboard

	Assembly bolt	Deck tape	Adhesive	Deck	Rubber pad	Base plate	Assembly nut	Pivot cup	Top washer	Top bushing	Bottom bushing	Bottom washer	King pin	King pin nut	Lower hanger	Axel	Wheel washer (inner)	Bearing (inner)	Wheel spacer	Wheel	Bearing (outer)	Wheel washer (outer)	Axel nut
Assembly bolt	x																						
Deck tape	w	x																					
Adhesive	w	t	x																				
Deck	w		t	x																			
Rubber pad	w			t	x																		
Base plate	w				t	x																	
Assembly nut	w					t	x																
Pivot cup						t		x															
Top washer						t			x														
Top bushing									t	x													
Bottom bushing											x												
Bottom washer											t	x											
King pin						s			s	s	s	s	x										
King pin nut												t	w	x									
Lower hanger								t		t	t				x								
Axel															t	x							
Wheel washer (inner)																w	x						
Bearing (inner)																w	t	x					
Wheel spacer																w		t	x				
Wheel																w		w	w	x			
Bearing (outer)																w			t	s	x		
Wheel washer (outer)																w					t	x	
Axel nut																w						t	x

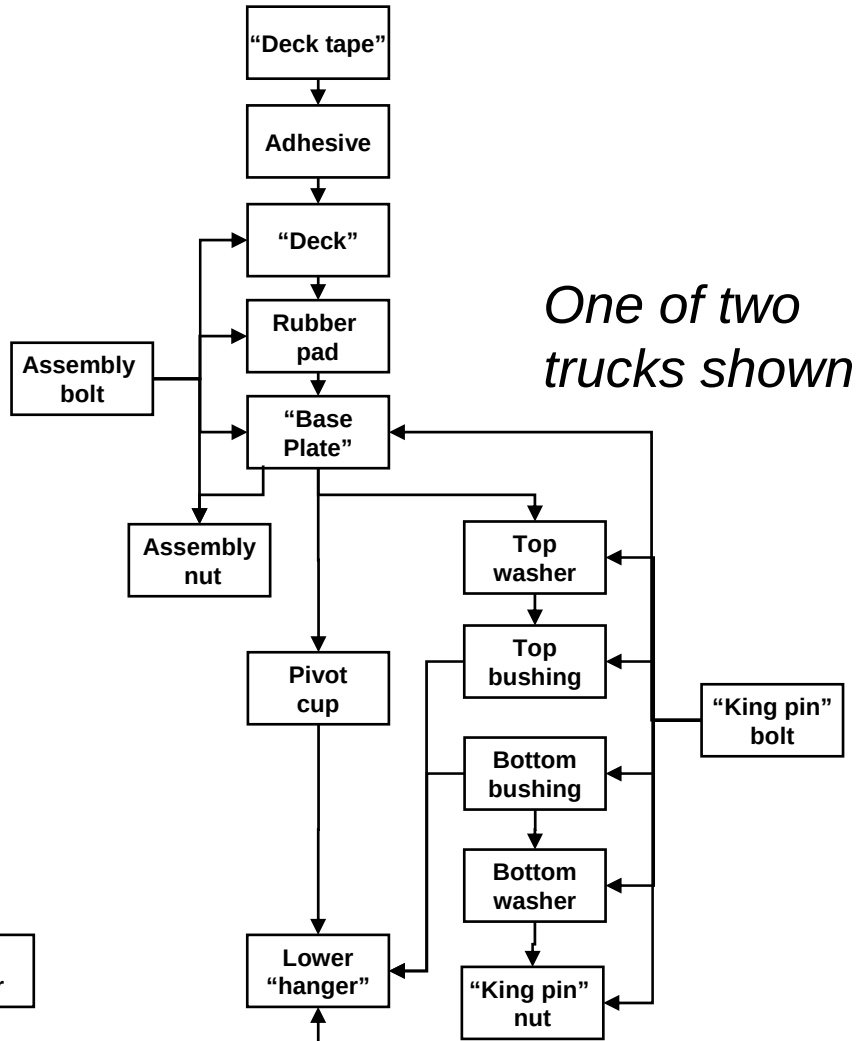
t = touches or tangent
w = within
s = surrounds

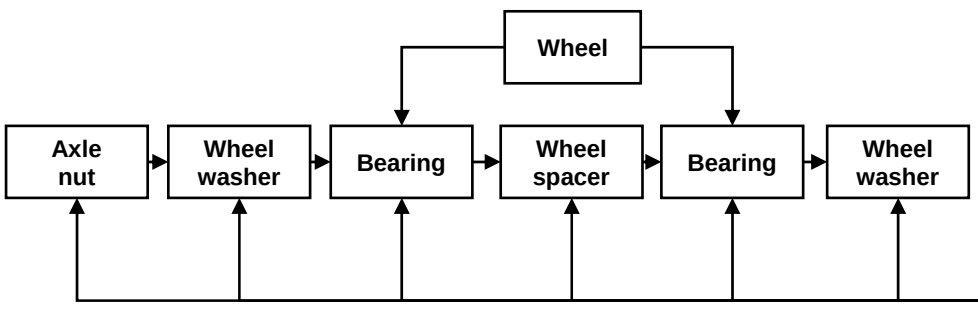
Implementation Structure

- **The implementation structure captures information about how the item was implemented**
- **Elements were linked by some manufacturing/assembly/ integration steps**
- **Examples:**
  - **Bonded to (I.e. was bonded)**
  - **Bolted to (I.e. was bolted)**
  - **Compiled with**
  - **Pressed against**
  - **Etc.**
- **These are issues of form**
- **Can also represent with object-object links in OPM**

Implementation Structure - Skateboard

- **Often very similar to, but not identical to the topological structure, but different information on links**
- **In a complete description, these arrows would be labeled as well**





Structure - "List" - Skateboard

- **Lower triangle contains the spatial structure, which would be symmetric**
- **Upper triangle contains the implementation structure, which would be symmetric**
- **This is starting to look like an architecture!**

	Assembly bolt	Deck tape	Adhesive	Deck	Rubber pad	Base plate	Assembly nut	Pivot cup	Top washer	Top bushing	Bottom bushing	Bottom washer	King pin	King pin nut	Lower hanger	Axel	Wheel washer (inner)	Bearing (inner)	Wheel spacer	Wheel	Bearing (outer)	Wheel washer (outer)	Axel nut
Assembly bolt	x			b	b	b	s																
Deck tape	w	x	g																				
Adhesive	w	t	x	g																			
Deck	w		t	x	p																		
Rubber pad	w			t	x	p																	
Base plate	w				t	x	p	p	p				b										
Assembly nut	w					t	x																
Pivot cup						t		x							p								
Top washer						t			x	p			b										
Top bushing									t	x			b		p								
Bottom bushing											x	p	b		p								
Bottom washer											t	x	b	p									
King pin						s			s	s	s	s	x	s									
King pin nut												t	w	x									
Lower hanger								t		t	t				x	st							
Axel															t	x	st	st	st		st	st	s
Wheel washer (inner)																w	x	p					
Bearing (inner)																w	t	x	p	p			
Wheel spacer																w		t	x				
Wheel																w		w	w	x	p		
Bearing (outer)																w			t	s	x	p	
Wheel washer (outer)																w					t	x	p
Axel nut																w						t	x

t = touches or tangent
w = within
s = surrounds

g = glued
b = bolted
s = screwwed
p = pressed
st = stacked

How do Form and Function Connect?

- **We have a high level model of the internal value related function of a skateboard**
- **And a detailed parts list, and understanding of the formal structure (in this case absolutely complete, with every part enumerated, not the usual case!)**

- **How do the elements and their structure allow the higher level value related externally delivered function to emerge?**

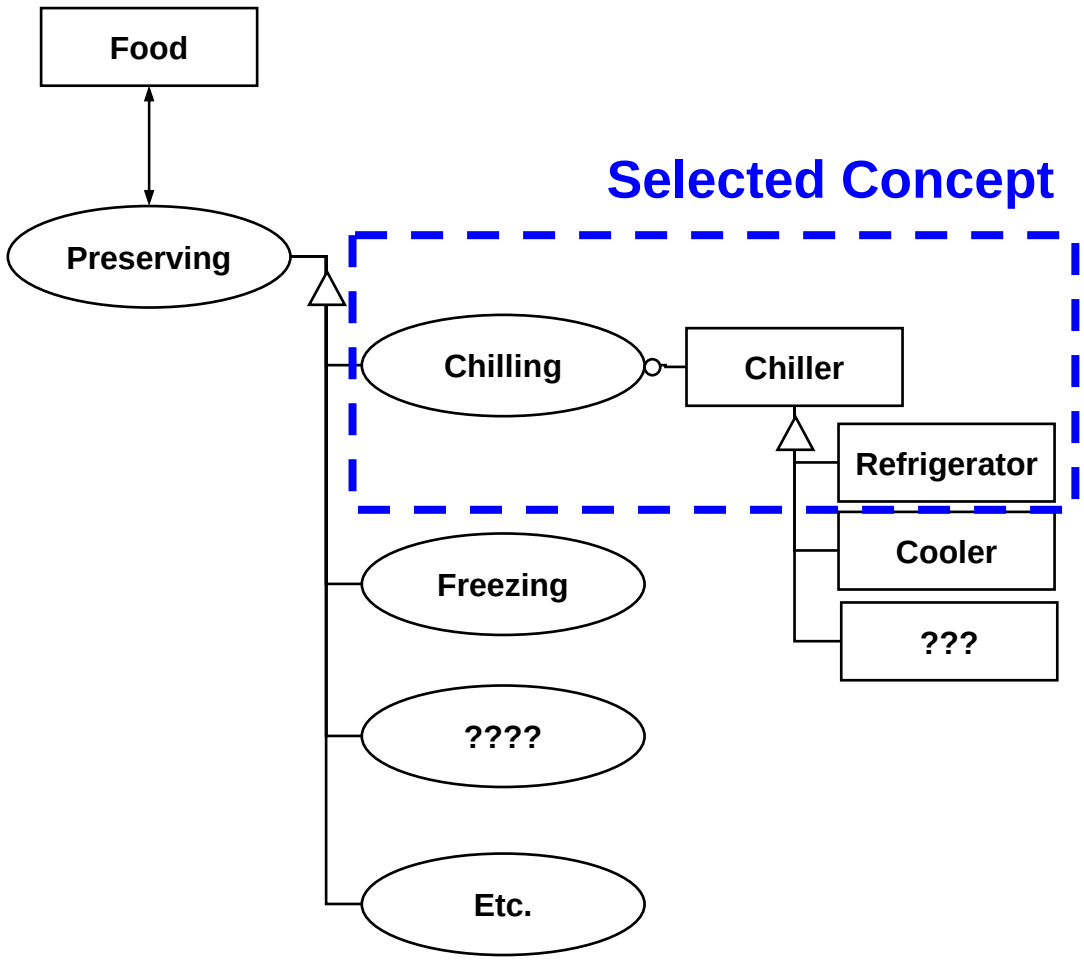
Note that we are reasoning "outer in"

Example - Refrigerator

- More integrated product/system - mechanical, electrical, thermal processes
- Really more complex than a medium system (about 300 parts of 200 types), but can be simplified to represent a medium system
- Model of physical/thermal process system

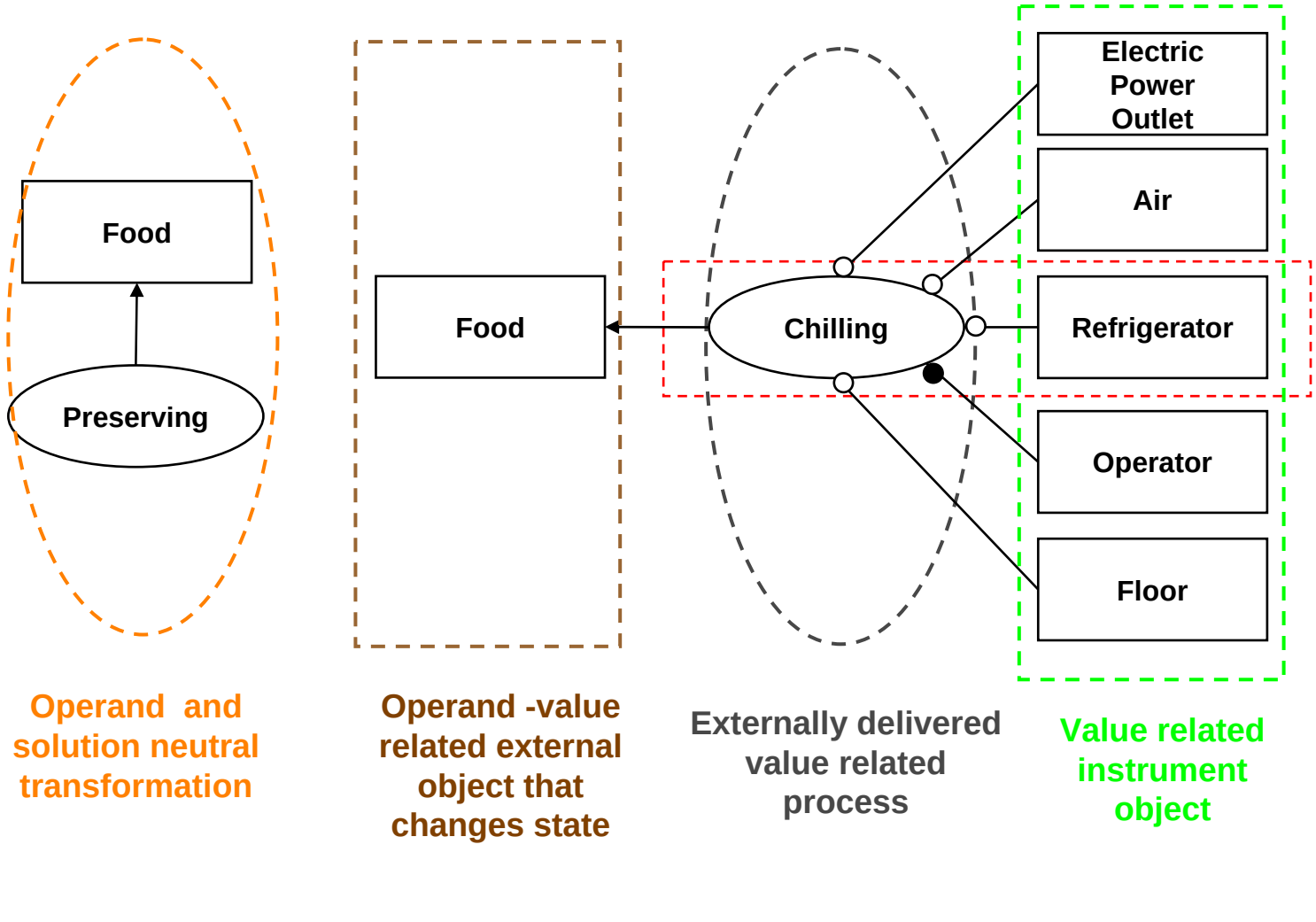
Concepts - Preserving Food

- **Solution neutral statement is: ‘preserving food’**
- **Solution specific processes: chilling, freezing, etc.**
- **Solution specific form for chilling: refrigerator, cooler, etc.**
- **Selected concept is chilling with a refrigerator**



▲ Decomposes to
△ Specializes to
◭ Has attribute of

Preserving Food Concept - Refrigerator



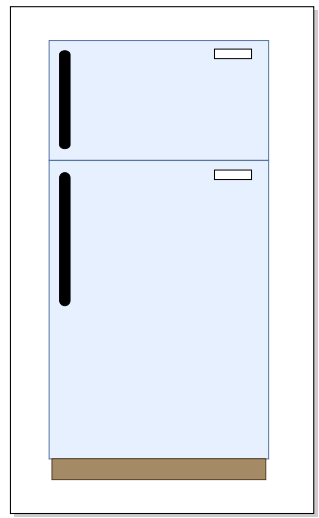
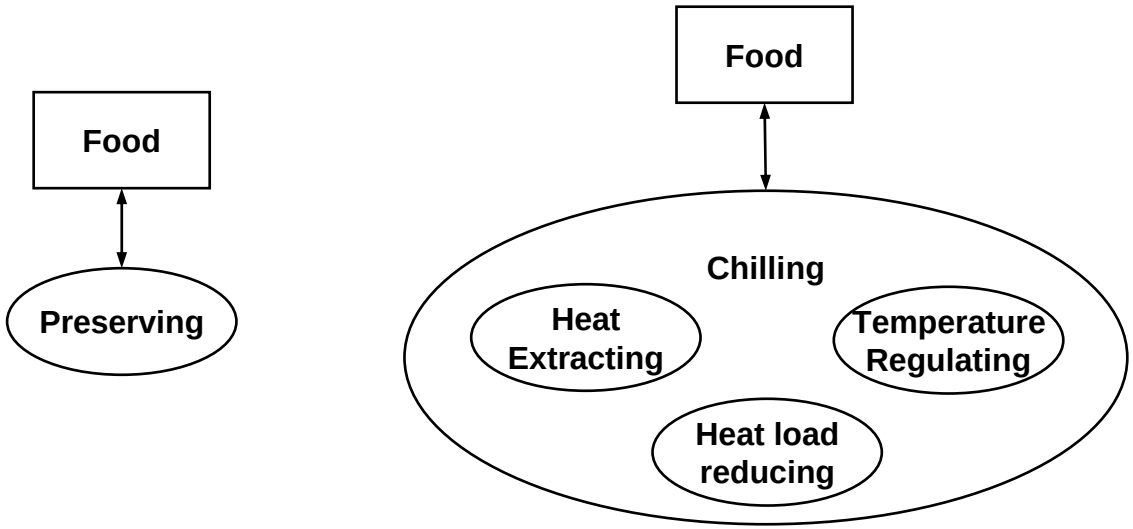


Figure by MIT OCW.

Multi-function Concepts for Chilling



▲ Decomposes to
△ Specializes to
◭ Has attribute of

"Chilling" implies cooling, but at a relatively constant temperature above freezing, and hence temperature regulating. Chilling efficiently implies that the ambient heat load on the process be reduced.

Refrigerator - Idealized internal value related processes

- Idealized internal value related processes and operands informed by the concept refrigerator

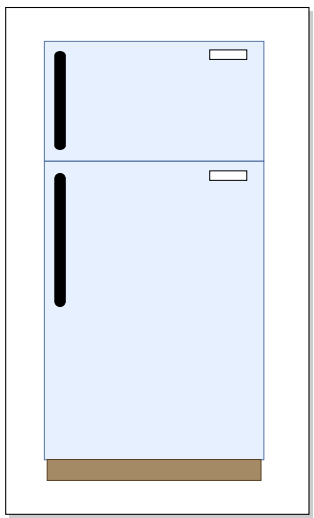
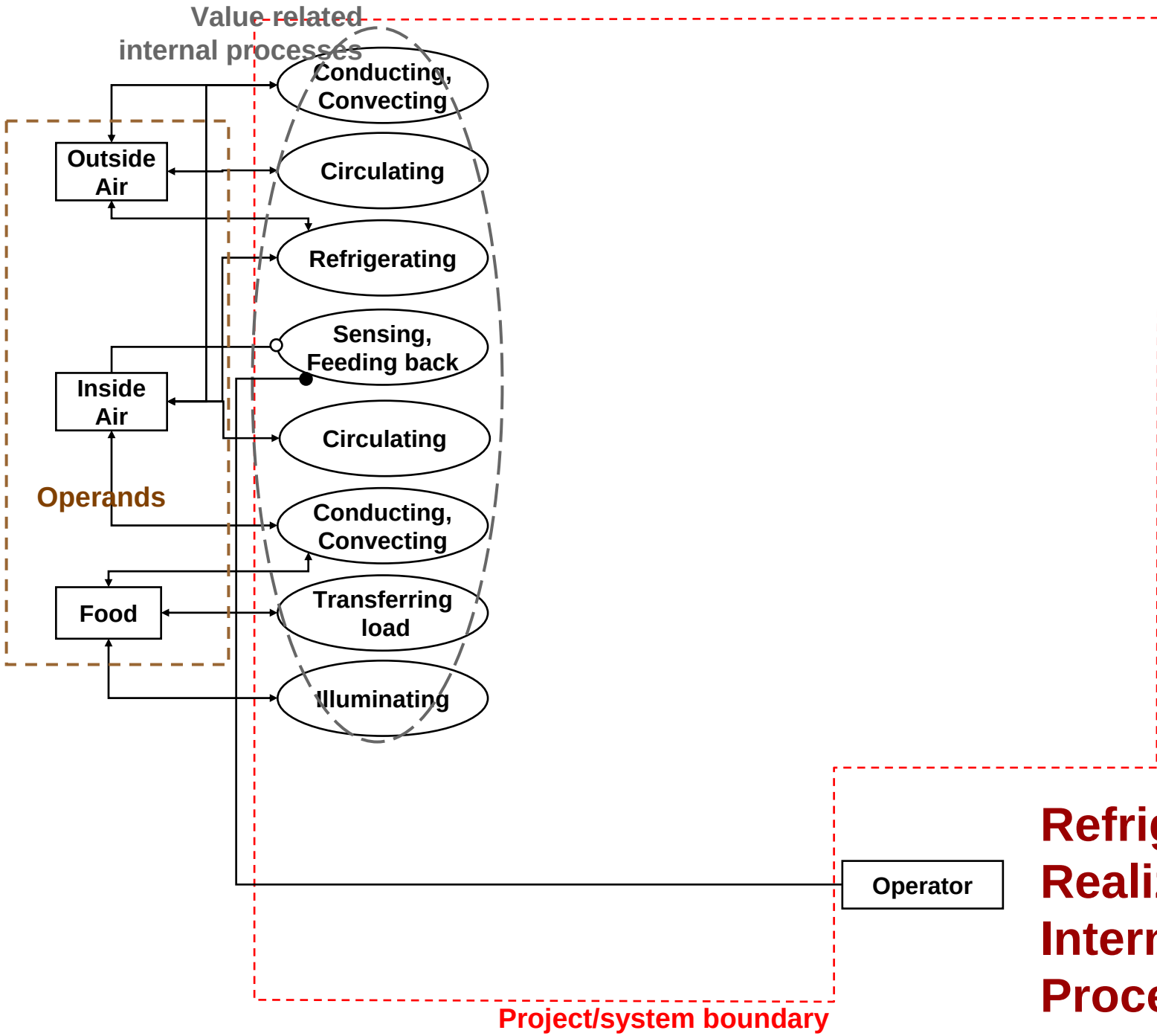


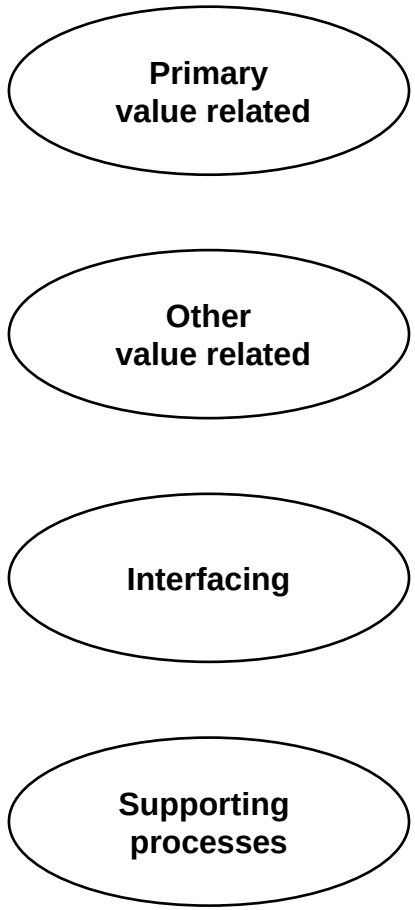
Figure by MIT OCW.



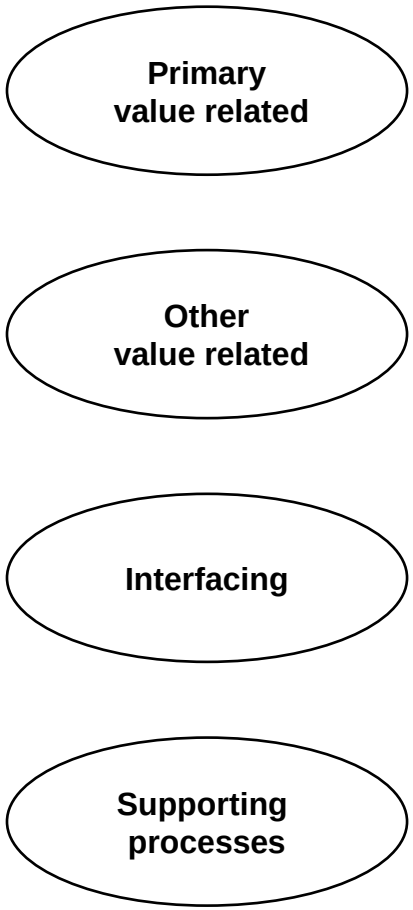
Refrigerator - Realizable Internal Processes

Primary Value, Other Value, Interfacing, and Supporting Internal Functions

- All product/systems have a primary value related process - why the product was purchased
- Many have other processes that deliver other or additional value, e.g. music in car, ice maker in frig
- All product/system have interface processes with the operands, other elements of the whole product system
- Most product/systems have other internal processes that somehow support the value processes, but do not them selves add any value



Primary Value, Other Value, Interfacing, and Supporting Internal Functions - Refrigerator



- All product/systems have a primary value related process - chilling food
- Many have other value related processes - ice making, dispensing cold water, freezing, ?
- All have interface processes - with food, floor, ?
- Most have other internal processes that support the value processes - structurally supporting, ?

Form of A Simple Refrigerator - List

evaporator fan, motor	compressor wiring harness	dairly compartment assembly	cabinet shelf ladder (l,r) support
evaporator shroud	running capacitor	door shelf assembly (3)	glass shelf assembly
wiring harness	compressor mount	door gasket	shelf assembly with track for basket
heat exchanger	starting relay	door trim	roll-out basket assembly
accumulator	overload protector	door pannel	crisper roller (l,r,l,r)
evaoporator coil	fan bracket (condensor fan)	door handle	crisper slide (l,r,l,r)
drain tube	control knob and indicator	door	center crisper assembly
drain trought assembly	controller (refig temp)	switch depressor	crisper tray assembly
drain pan	control bracket	light diffuser	crisper glass assembly
condenser	light bulb (4)	fan guard	criper draw assembly
fan switch	light stand off	door hinge (top)	louvered grille
light switch	light socket	door hinge (bottom)	compressor fan shroud assembly
switch housing	control light and socket	door frame (top, sides)	compressor shroud assembly
condensor fan, schroud, motor	power cord	back cover	control pannel
condensor schroud	light terminator	legs, rollers	evaporator cover
compressor	egg tray	base assembly	cabinet assembly
condensate heater loop		kickplate	

- **Parts list for a simple refrigerator, no ice maker, cold water dispenser, freezer, etc.**
- **66 part types in list is already simplified**
- **Actually about 210 part numbers on bill of material**

Rationalize Element List

	primary value elements and assembiies		elements in assemblies or **important details in element**	supporting and secondary element, connectors, etc.	other value related elements
cabinet assembly	cabinet assembly		**cabinet, insulation**	**sturcture**	**outer pannels**
light diffuser				light diffuser	
fan guard				fan guard	
door hinge (top)	hinges		door hinge assembly (top)		
door hinge (bottom)			door hinge assembly (bottom)		
door frame (top, sides)				door frame (top, sides)	
back cover				back cover	
legs, rollers	legs, rollers		legs, rollers		
base assembly				base assembly (beams)	
kickplate				kickplate	

- **Try to rationalize element listing to a more manageable number 20-40**
- **Some important elements that are highly integral will have to be *expanded* - e.g. cabinet assembly to cabinet, insulation, structure, outer panels**
- **Some elements can be grouped into abstractions - e.g. top and bottom door hinges to hinge**
- **Some can be identified as being associated with supporting or secondary elements or connectors - e.g. light diffuser, fan guard, base assembly beams**
- **Some can be identified with other value functions - e.g. outer panels**

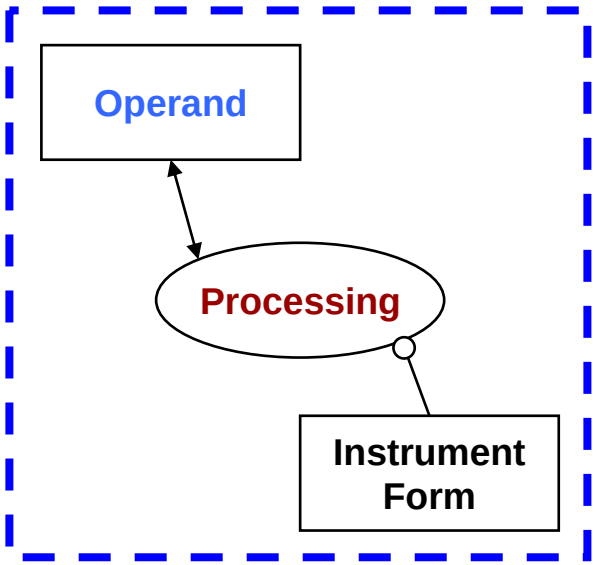
How do Form and Function Connect?

- **We have a high level model of the internal value related function of a skateboard**
- **And a detailed parts list, and understanding of the formal structure (in this case absolutely complete, with every part enumerated, not the usual case!)**

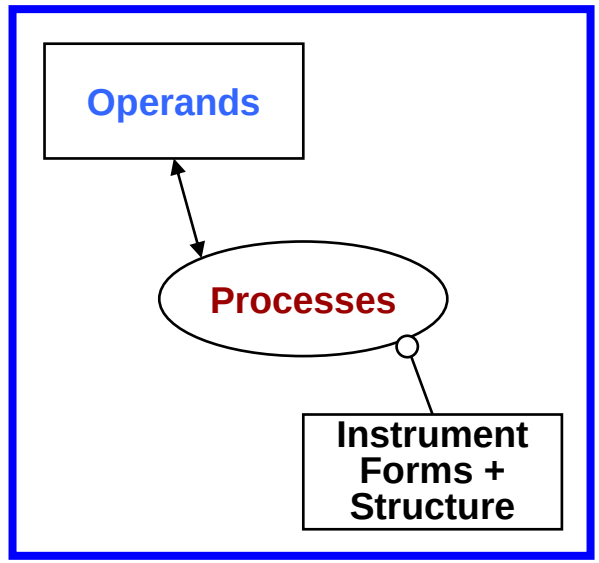
- **How do the elements and their structure allow the higher level value related function to emerge?**
- **Is their evidence of interfacing functions?**
- **Is their evidence of value related functions other than the primary one?**
- **Is their evidence of internal "supporting functions" other than primary functions**

Note that we are reasoning "outer in"

The Product/System Architecture

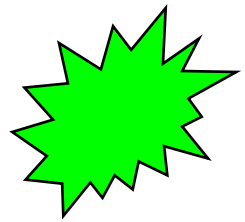


Concept



Architecture

Synthesized PDP



Group reports on:

- **Steps which appeared in most or all PDP's**
- **Synthesis into reference PDP**
- **Distinguishing features**

Closure on Definitions

- System
- Complexity
  - Dynamic
  - Large
  - Perceived Complexity
- Part
- Detail
- Atomic Parts
- Product
- Value
- Benefit
- Product/system

Closure on Definitions

System

- **A set of interrelated elements which perform a function, whose functionality is greater than the sum of the parts [Reference]**

Alternate definitions

- **Two or more elements that interact by design or coincidence**
- **Interacting parts or elements that can be regarded as a whole (within a boundary)**

Closure on Definitions

Complexity

- having many interrelated elements and interfaces [Reference]

Complex Systems

- have many levels of elements, types of elements, connections and types of connections
- require a great deal of information to specify

Related Concepts:

Evolving (process)

- having evolving requirements or resources

Large (team)

- requiring a team larger than one which can communicate directly among themselves

Complicated

- appearing to the observer as being difficult to understand (an issue of perception)

Closure on Definitions

Product	• **A thing which can be delivered or transferred and has value**
Value	• **Benefit at cost**
Benefit	• **Worth, importance, utility as judged by a subjective observer (the beneficiary)**
Product/ system	• **A product which is also a system, emphasizing the dual nature**

Closure on Definitions

Part

- **A part is an element that you cannot take apart and then reconstitute in its original form - it has been irreversibly implemented [no link to function],** ***or***
- **A part is an element that you cannot take apart without destroying its ability to deliver its function [explicit link to function]**

Detail

- **An element of a part (so a part can be a system)**

Atomic part

- **A part,** ***or***
- **The details of a part which have independent function**

Additional material on fundamental processes - FYI

Fundamental Processes

A la Crawley

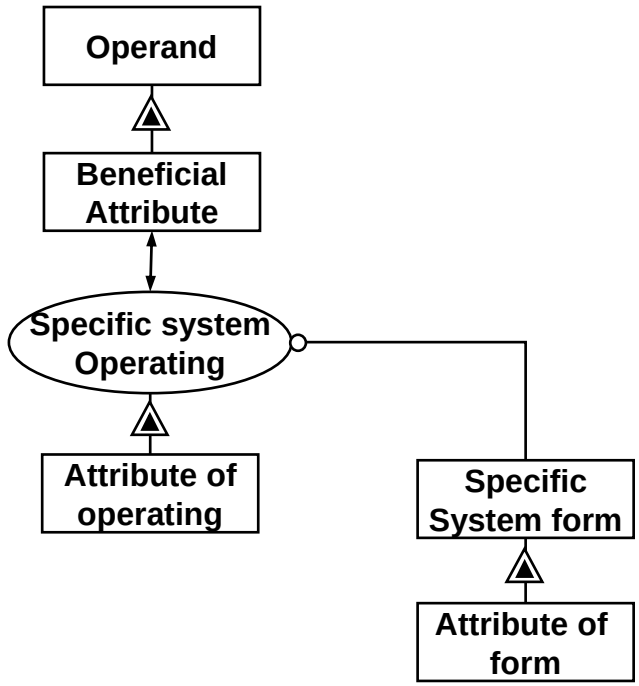
- **Create (and Destroy)**
- **Transport**
  - **In place - from A to B, or to "spatial storage" and recover from "storage"**
  - **In time – only delays allowed since time is causal "temporal storage"**
- **Transform**
  - **In type or form**
  - **In quantity – magnitude for continuous attributes, number for discrete artefacts**
- **Compare**
  - **Any of the place, time, type or quantity [not sure it is independent of Transform]**

Fundamental Process Frameworks

Dori	**Create, Destroy**	**Transform**					**-**
Crawley	**Create, Destroy**	**Transport**		**Transform**			**Compare**
		Place	**Time (delay)**	**Type/ Form**	**Quantity**		
					Magnitude (continuous)	**Number (discrete)**	
Pahl & Beitz	**-**	**Place (channel)**	**Time (store)**	**Type (change)**	**Magnitude (vary)**	**Number (connect)**	
Turing	**Create**	**Move**	**Store**	**Read, write**	**-**	**Write**	**Look up (compare and locate)**
Bool	**-**	**-**		**-**	**-**	**And, Or**	**(Equivalence)**

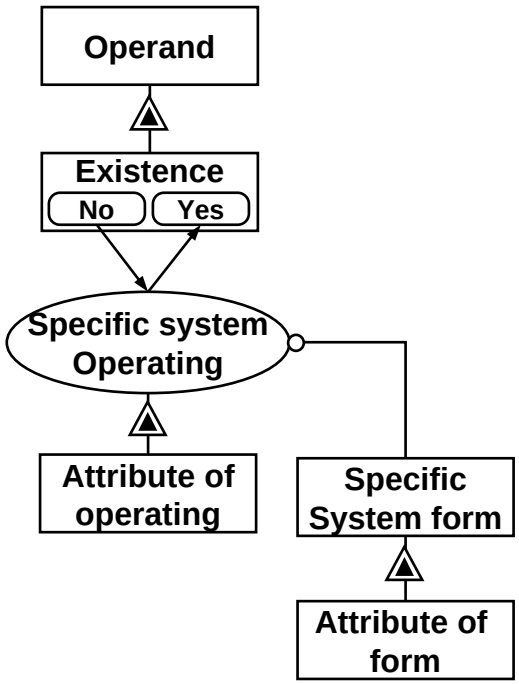
Map fundamental processes to OPM

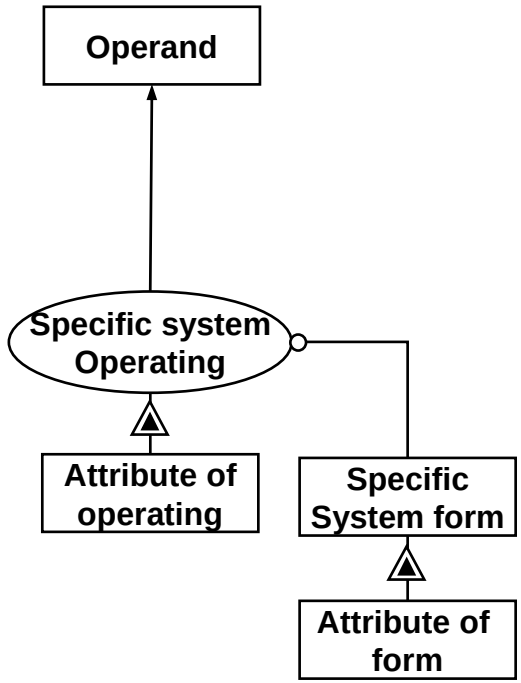
- Try to map fundamental processes on generic OPM of system operating
- See if this leads to any systematization of the classical fundamental processes
- Explore if this forms a basis of predicting emergence or idealized system design



Creation/Destroy

- **Can show existence explicitly or implicitly**
- **Have to be cautious as to what is really *created* - an object (I.e. an arrangement of things) or its mass**
- **Has to do with a fundamental change in the existence of something**

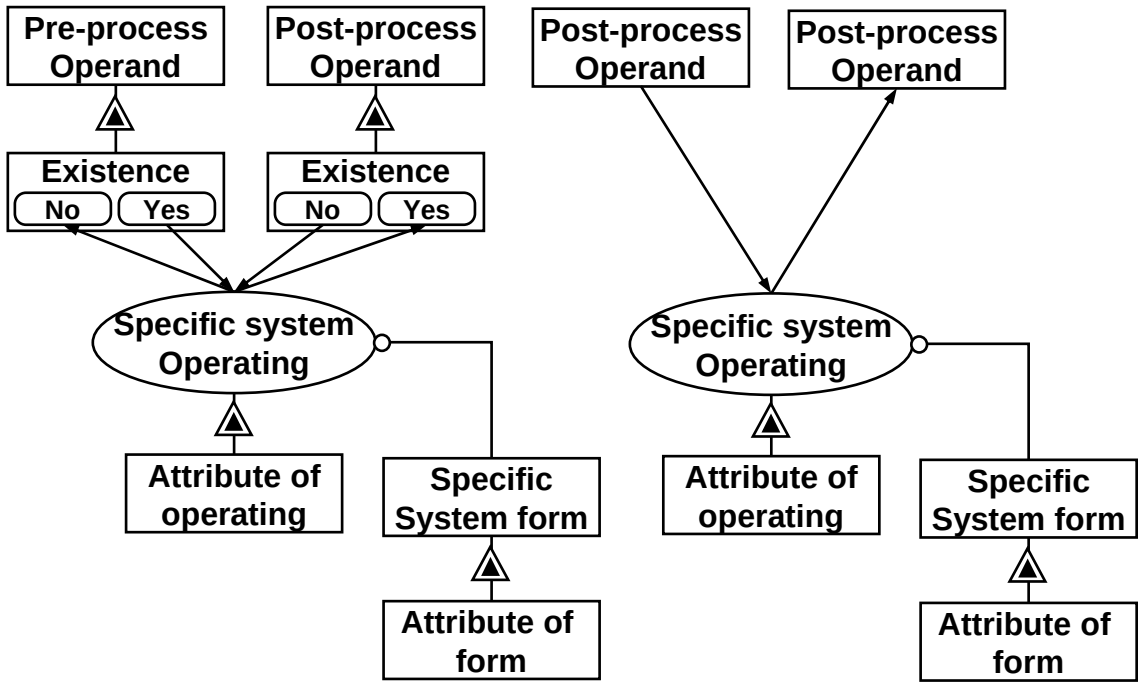




Destroy has arrows in opposite sense

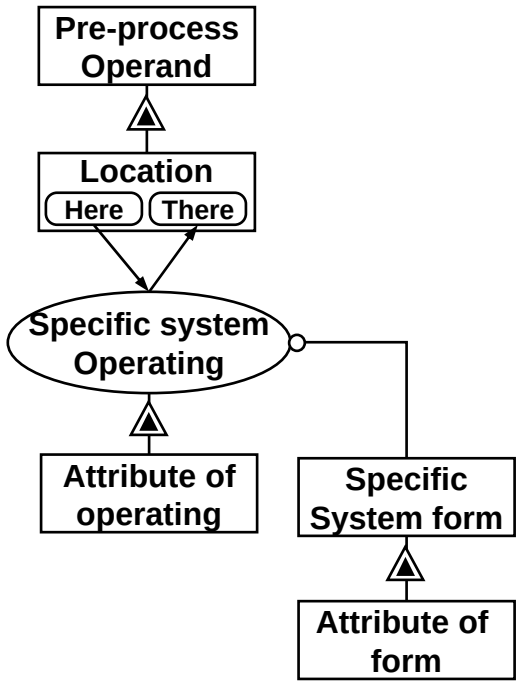
Transform Type/Form (Change)

- Can show existence explicitly or implicitly
- Has to do with a fundamental change in the existence of something
- Something is destroyed and something entirely new is created



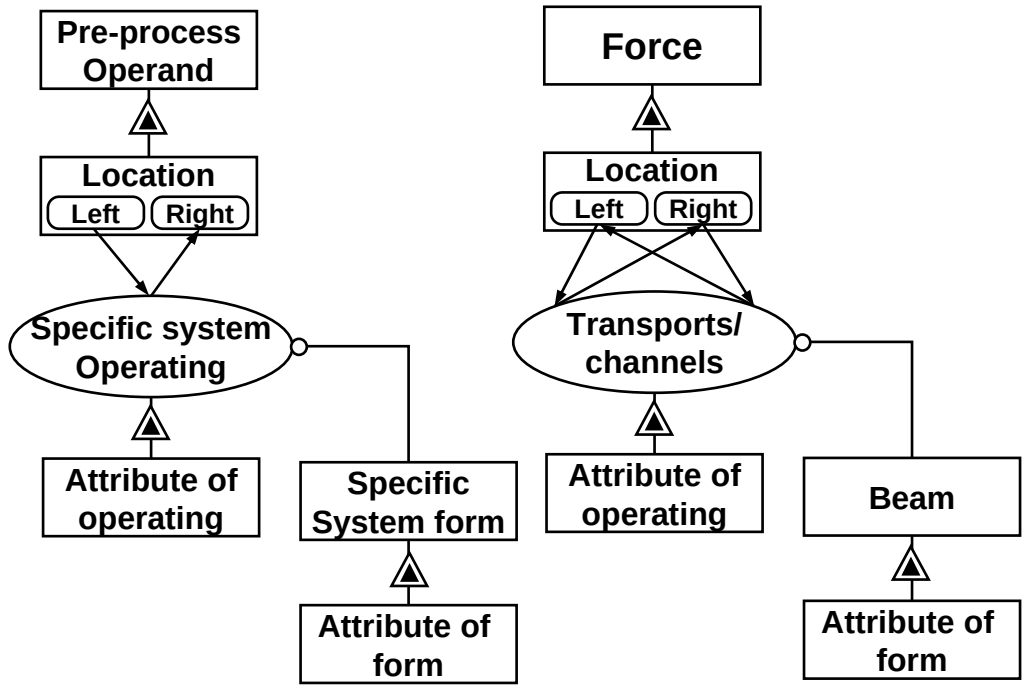
Transport (discrete object)

- **Has to do with a fundamental change in location of something**
- **Changes the location attribute of the operand**
- **Must now use attribute that changes (vs.. creation)**

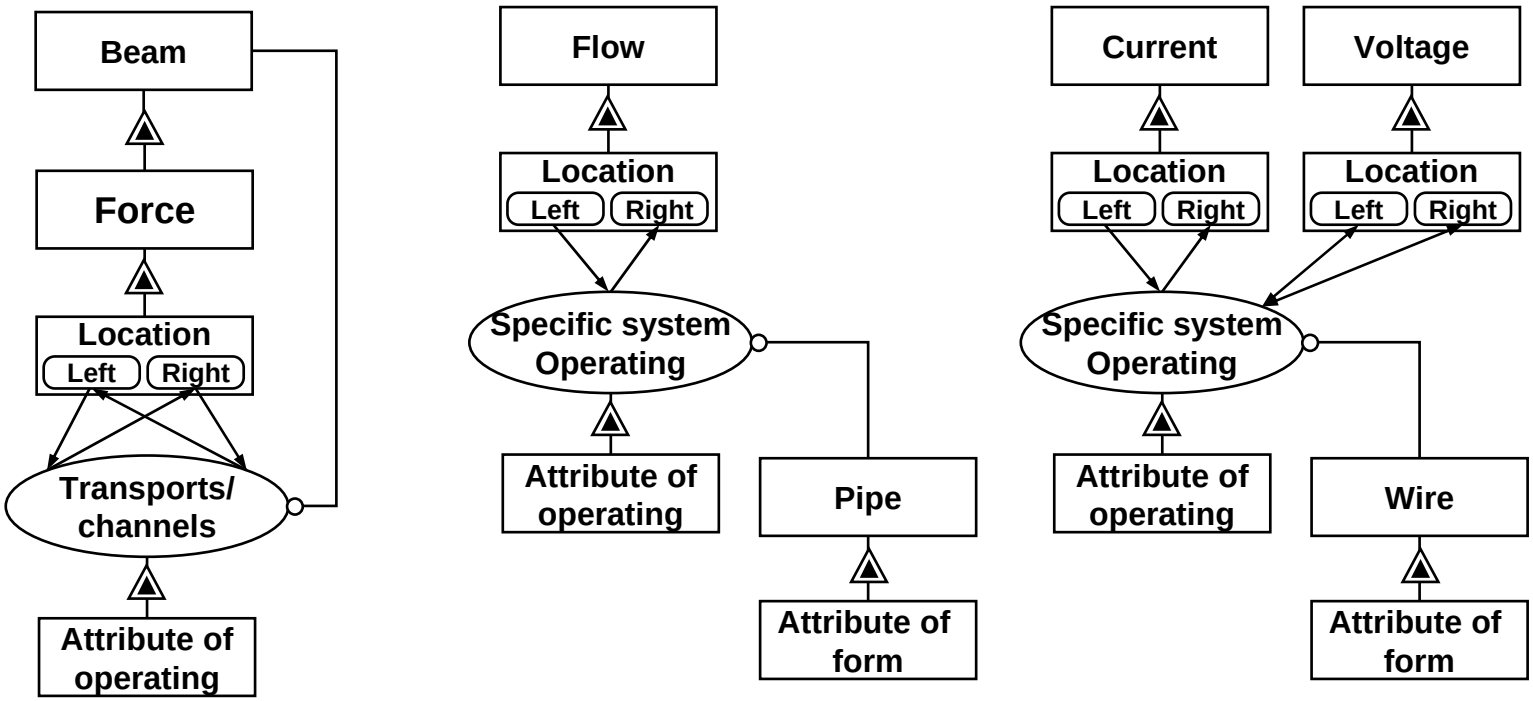


Transport/channel (field variable)

- **Has to do with a fundamental change in location of something, but that is more of a field variable like heat, stress, charge**
- **Changes the location attribute of the operand, but may be bi-directional**



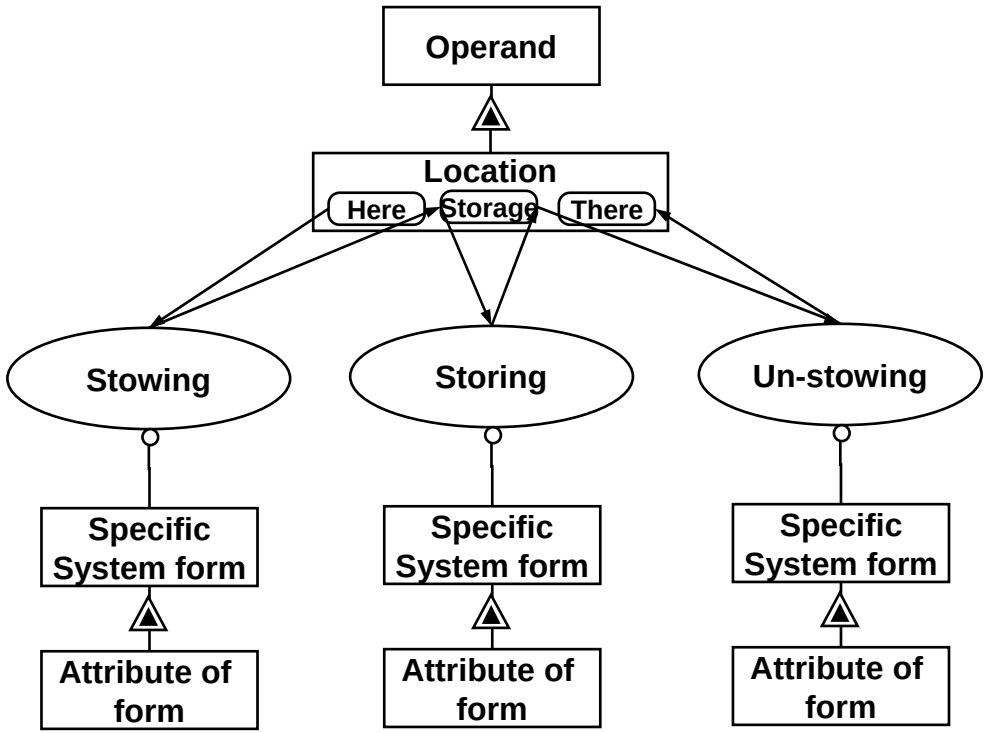
Transport/channel (field variable)



Alternate

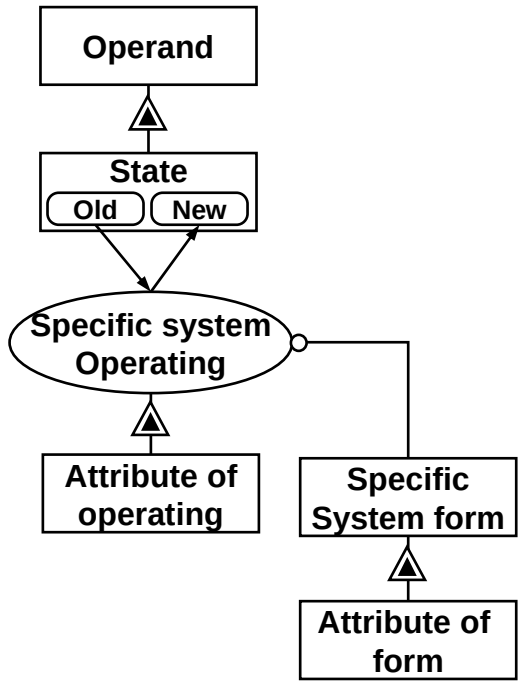
Store

- **Complex idea of moving something to a storage location, leaving it there some latency time, and then recovering**
- **Storing process may include or be in addition to the stowing and un-stowing processes**
- **Field variable also store, e.g. energy storage in an electric field**



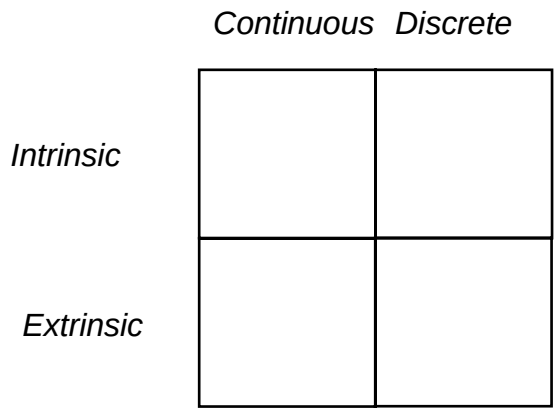
Vary (Magnitude)

- Has to do with a fundamental change in a continuously variable state of an operand other than location (e.g. amplitude, temperature)
- Could you extend this to discrete states without loss of generality (e.g. color)
- Could extend to states that, like location, don't really change anything about the object itself (e.g. ownership)
- Why distinguish this from transport, which just changes a different state, associated with location?



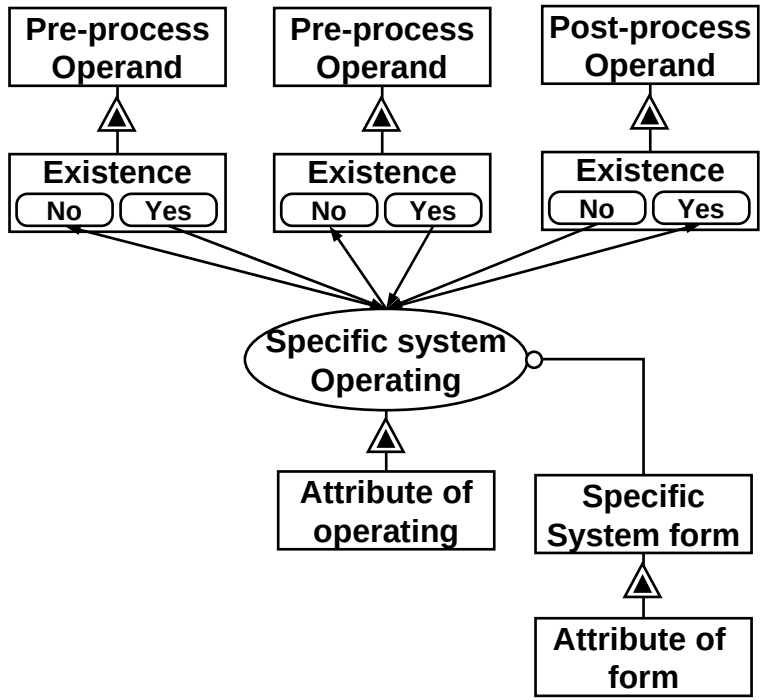
Classes of States

- **States that are continuously variable and actually effect the "intrinsic state" of the object**
  - **Temperature, pressure, voltage, current, etc.**
- **States that are discrete and actually effect the "intrinsic state" of the object**
  - **Color, size, material composition, on/off, in storage/not, created/not**
- **States that are continuously variable and do not actually effect the "state" of the object**
  - **Location**
- **States that are discrete and do not actually effect the "state" of the object**
  - **Ownership**



Transform Number (Connect)

- **This is in fact a class of processes that is probably richer than all the others combined**
- **Could include:**
  - **Connection of two objects to form a joint object (as in assembly)**
  - **Flowing together of two objects (fork in a river)**
  - **Combination of a physical object and an information object (as in a controlled process)**
  - **Processes conditional on the status of an object**
- **We have to expand this considerable, once we understand the one and two operand processes**



2 to 1 shown, could Also be 1 to 2

Issues raised in Mapping fundamental processes to OPM

- **Class of state: continuous, discrete, intrinsic, extrinsic**
- **Number of operands: one, two, three (more??)**
- **Nature of process (transport, store, vary, etc)**
- **Can we connect to notions of abstract algebra**