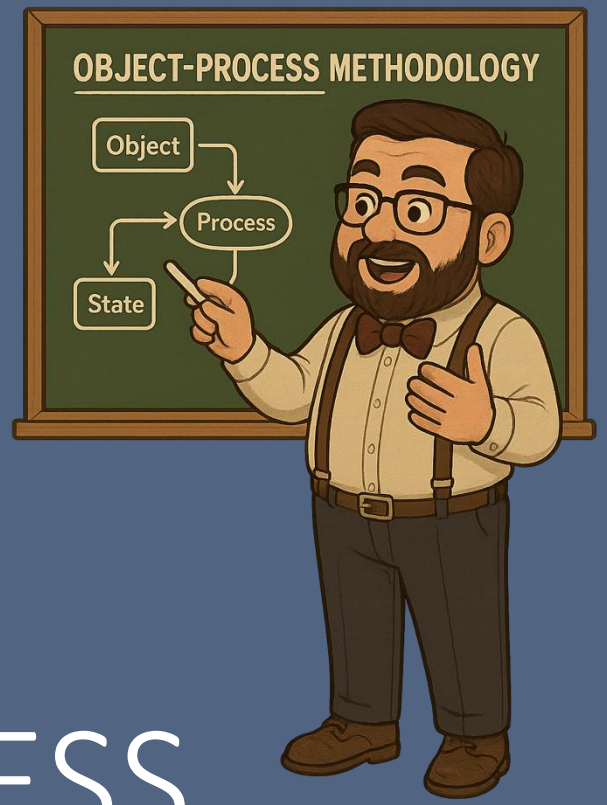
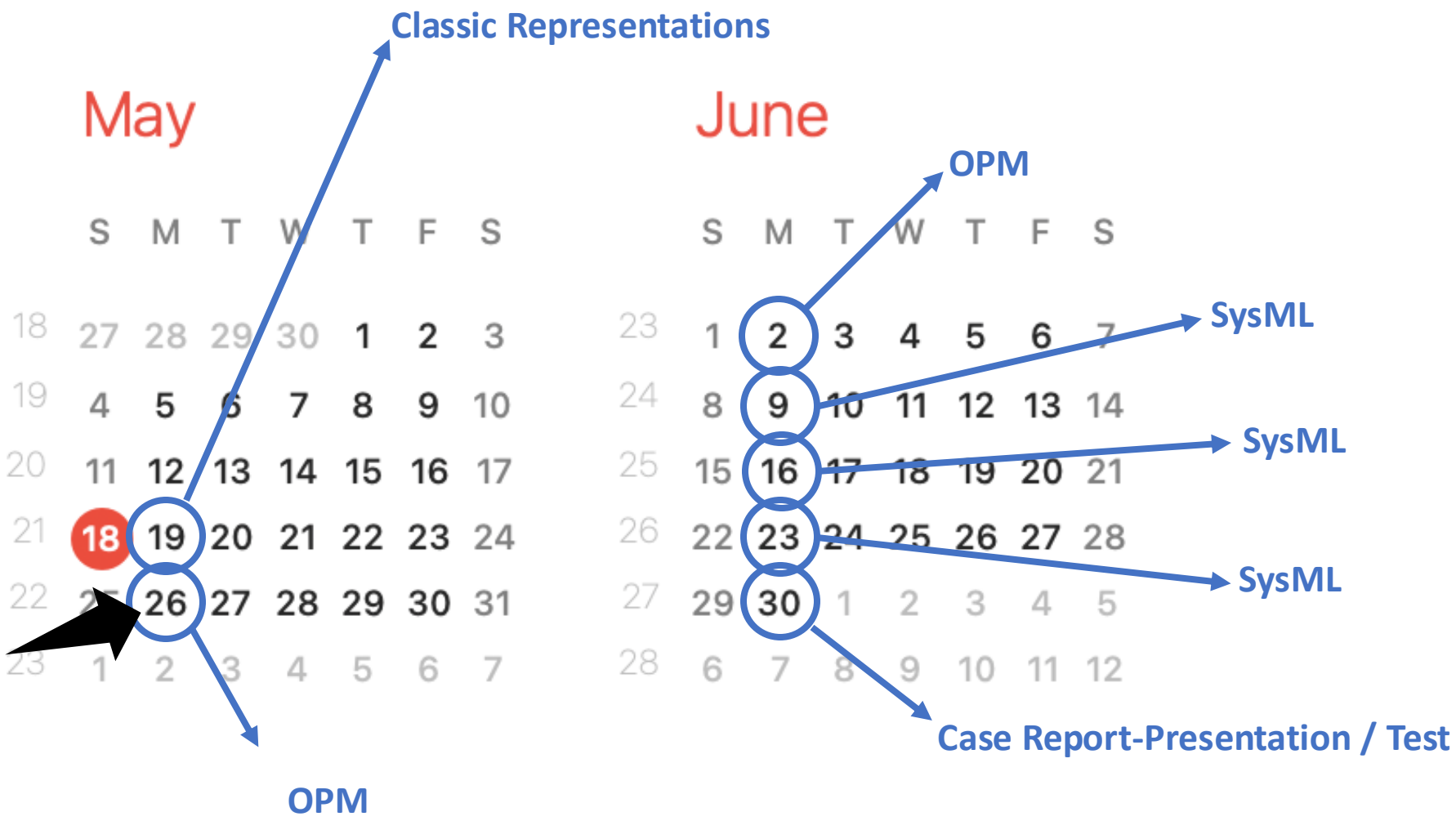
PEE-33




# OPM – OBJECT PROCESS METHODOLOGY

2025 – Prof. Dr. Christopher Shneider Cerqueira

# 6 Weeks...


Classic Representations
May
June
OPM
SysML
SysML
SysML
OPM
Case Report-Presentation / Test

# Learning Objectives

- 1 - Knowledge level:
  - L0-01 – Match Systems Engineering description to OPM
  - L0-02 – Remember OPM Basic symbols
- 2 - Apply level:
  - LO-03 – Use the OPM symbols
- 6 – Create level:
  - LO-04 – Create a concept model using OPM

→ **Reading Exercises**

→ **Hands-on Exercises**

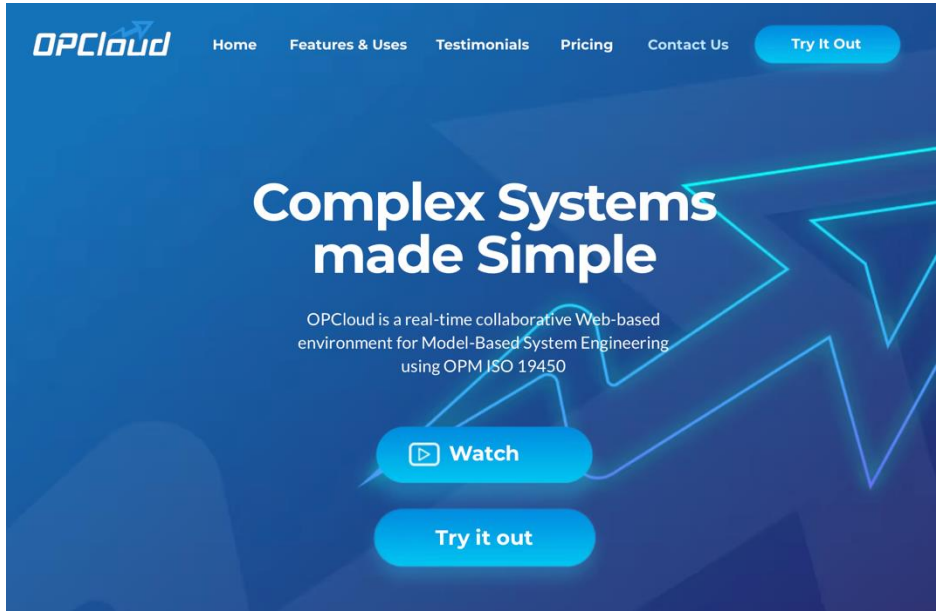
→ **Project Application**

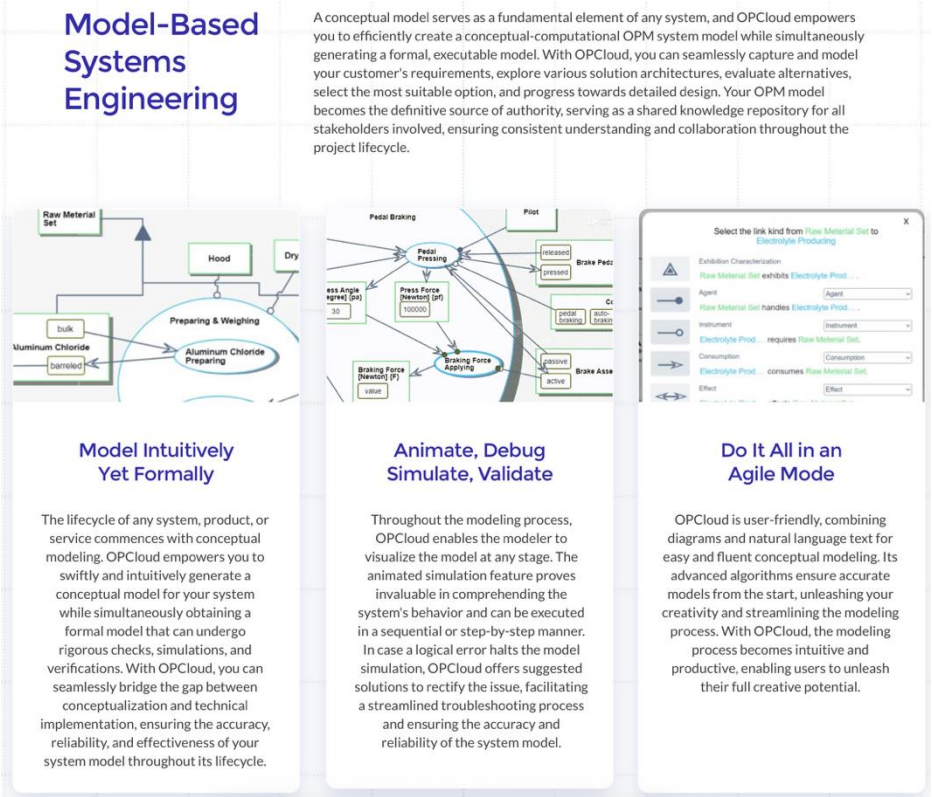
# Exercises of this lecture

- Reading Exercises:
  - 10 Questions about Dori's Book teaching the basic symbols and foundations.
- Hands-on Exercises: *joy
  - Two practicing lists from the book
- Project Application:
  - Model and simulate the concept of your project with a "single page" model.

# Resources for this lecture

- OPCloud
- **Logged account:**
  - https://opcloud.systems/
  - Connect w/ Google SSO using your **@ga account**
- **Not logged**
  - It **does not save** it, but works similarly
  - https://www.opcloud.tech



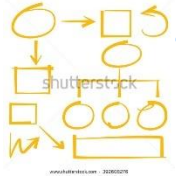



# Conceptual Modeling

# The start: How do we explain ideas to each other?

Grab a pen and piece of paper, or a chalk and blackboard

Scribble shapes with names next to them



While talking, run lines with or without arrows among the shapes

Follow the reaction of the audience to see if idea is understood

Answer questions, continue scribbling…

# The start: These “first” ideas → Conceptual Modeling

- A systematic, formalized process of describing, specifying, designing or explaining ideas, systems, products or processes through a model

- Applicable to both:
  - **Science** – Studying what is known and what is missing to satisfy human thirst for knowledge, and
  - **Engineering** – Designing systems to benefit humans, based on sound scientific principles

- Science can be thought of as reverse engineering of nature

# Conceptual Modelling

Convert — fragmented knowledge into explicit and integrated knowledge

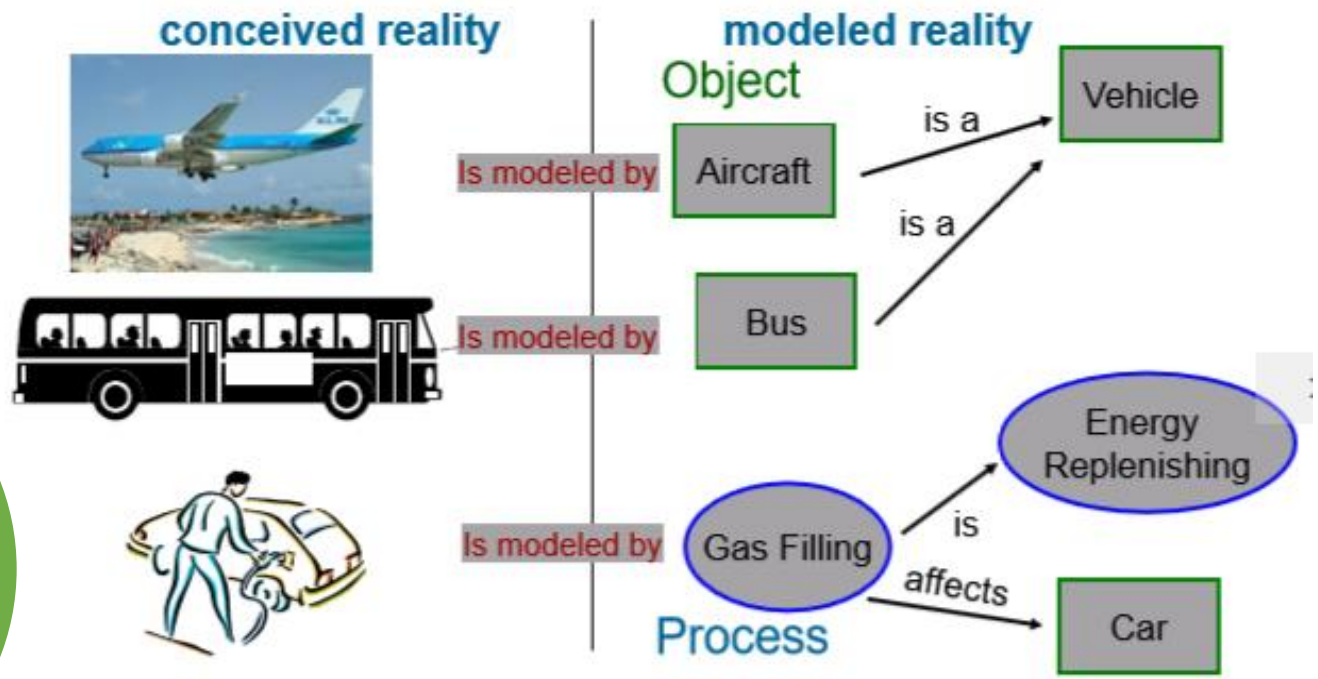
+

Construct — concise models

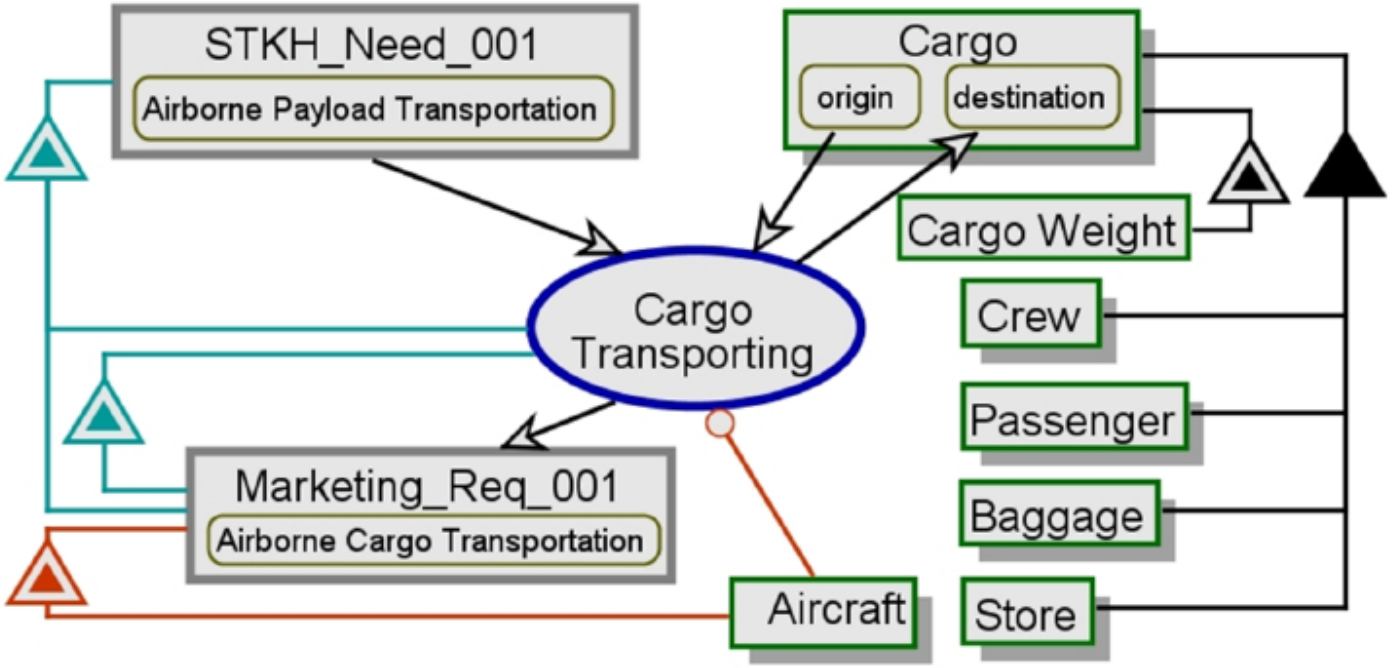
+

Communicate — The model to stakeholders

→ **Concept Models**




- **simple** yet **expressive**, and
- **intuitive** yet **formal**




STKH_Need_001 is Airborne Payload Transportation.
Cargo Transporting consumes STKH_Need_001.
Cargo Transporting yields Marketing_Req_001.
Marketing_Req_001 is Airborne Cargo Transportation.
Marketing_Req_001 exhibits Aircraft.
Cargo Transporting requires Aircraft.
Aircraft is physical.
Cargo Transporting changes Cargo from origin to destination.

Function Defining
Requirements Identifying
Requirements Allocating

STKH_Need_001 exhibits Marketing_Req_001, as well as Cargo Transporting.
Cargo Transporting exhibits Marketing_Req_001.

Traceability

Cargo is physical.
Cargo can be origin or destination.
Cargo exhibits Cargo Weight.
Cargo consists of Passenger, Baggage, Store, and Crew.
Passenger is physical.
Baggage is physical.
Store is physical.
Crew is physical.

Configuration management

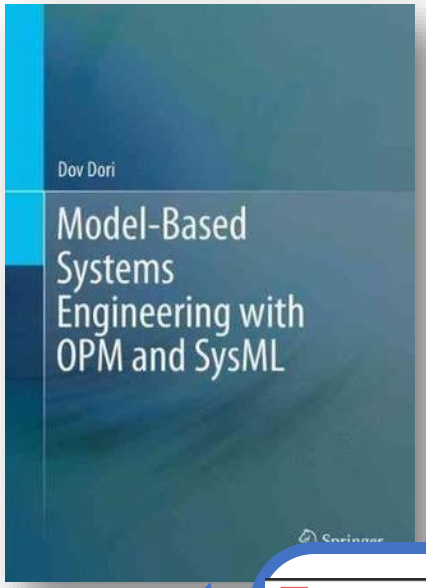
**Model-Based Systems Engineering for Aircraft Design with Dynamic Landing Constraints Using Object-Process Methodology ,** AIAA Scitech 2019 Forum

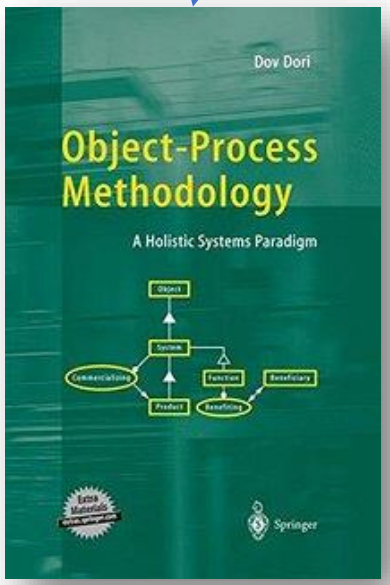
# OPM

Prof. Dov Dori

Created in 2002

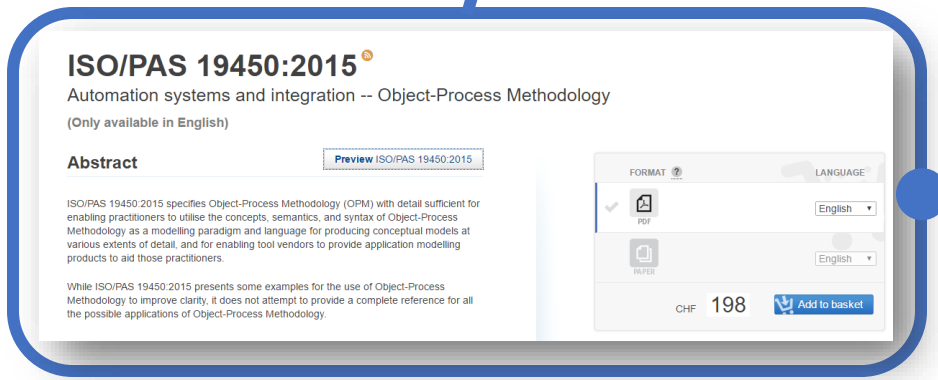
- **A** Single Diagram – Maps Behaviour and Structure
- **2 Building Blocks and 10 basic relations**
- Designed to “Systemic View” and “Concept Modelling”
- Simulation Ready

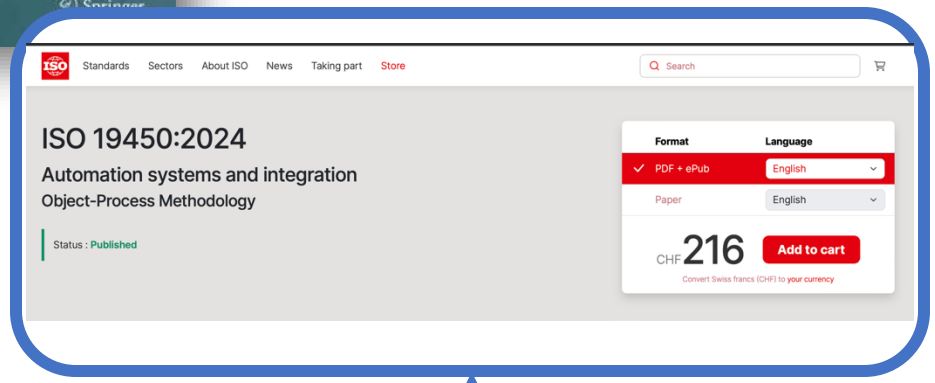



improving and showing it applicability

- ~130 Pages standard
- Published in late 2015
- Intended to *“Automation Systems and Integration”*
- Has the “**power**” of a ISO seal.




Added complexity and integrations

# I had the honor to met Prof. Dori in 2022

# References for this section

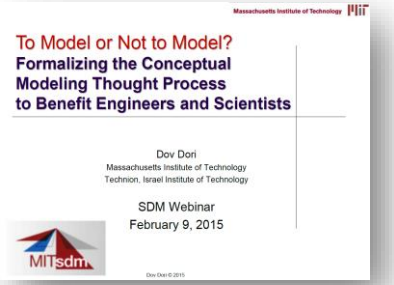
- Main:
  - DORI, D. Model-Based Systems Engineering with OPM and SysML. New York: Springer, 2016. ISBN 978-1-4939-3294-8.
- OPM Starting Guide
  - http://esml.iem.technion.ac.il/introduction-to-opm/
- Absurdly good:
  - https://en.wikipedia.org/wiki/Object_Process_Methodology



The following slides have “pieces” of the presentations/papers of the OPM Starting Guide

# OPM Foundations

# [Dori] Fundamental Questions:




- **1. What is needed to describe the universe?**
  - "Things" and their "relations"
- **2. What can those things do?**
  - Things can exist or happen.
- **3. What are the things that exist in the world?**
  - Objects exist – statics (time-independent).
- **4. What are the things that happen in the world?**
  - Processes happen – are dynamics (time-dependent).

# [Dori] Fundamental Questions:

- **5. How do objects and processes relate?**
  - Processes happen to objects. While happening,
  - Processes transform objets.
- **6. Transform?? what does that mean?**
  - Create
  - Destroy
  - Affect
    - an object

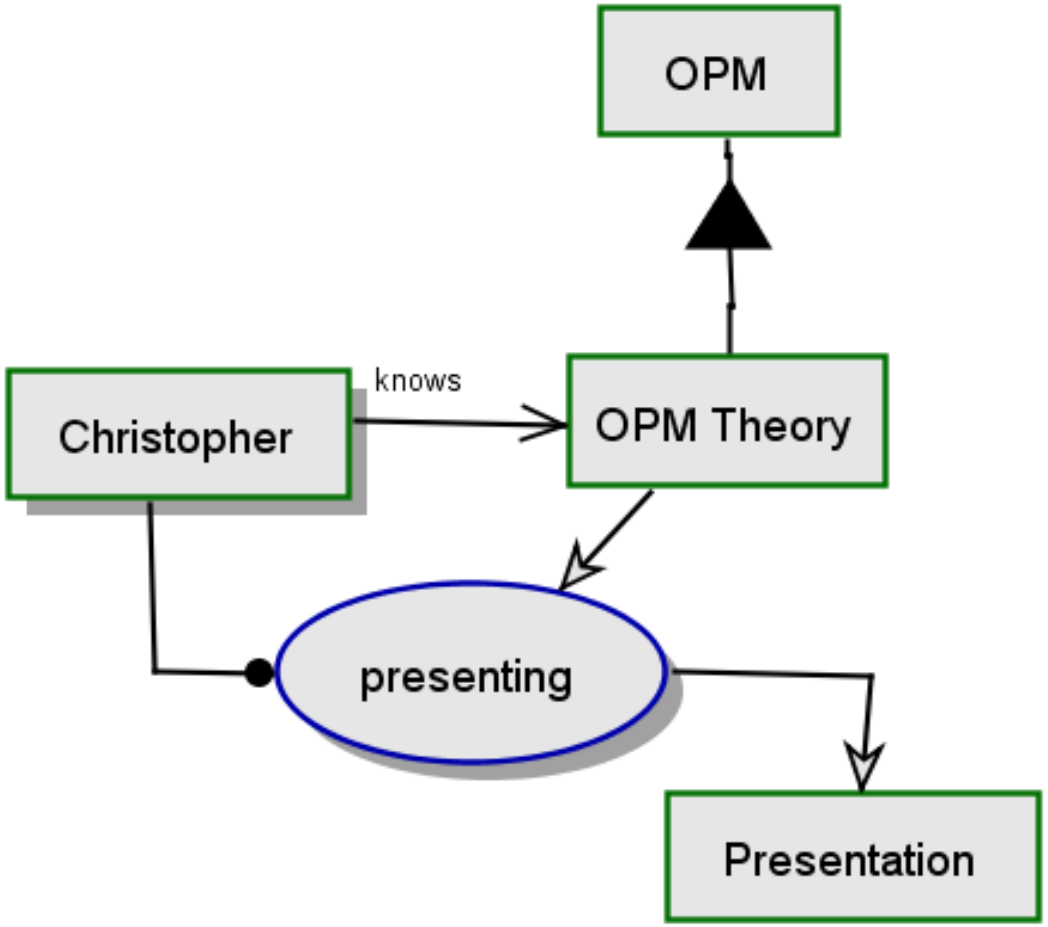
# [Dori] Fundamental Questions:

- **7. Affecting? What does that mean?**
  - A process affects an object by changing its state. Hence, objects must have states.
- **8. What are the two major aspects of any system?**
  - Structure: static aspect – what the system is made of?
  - Behaviour: dynamic aspect - how the system changes over time?
- **9. Which third aspect is specific to man-made systems?**
  - Function: the utilitarian, subjective aspect. Why? for whom? Who benefits?

# [Dori] Object-Process Theorem

**Objects** with **states**, **processes** and their relations among them constitute a **necessary** and **sufficient universal ontology** to describe a **system**.

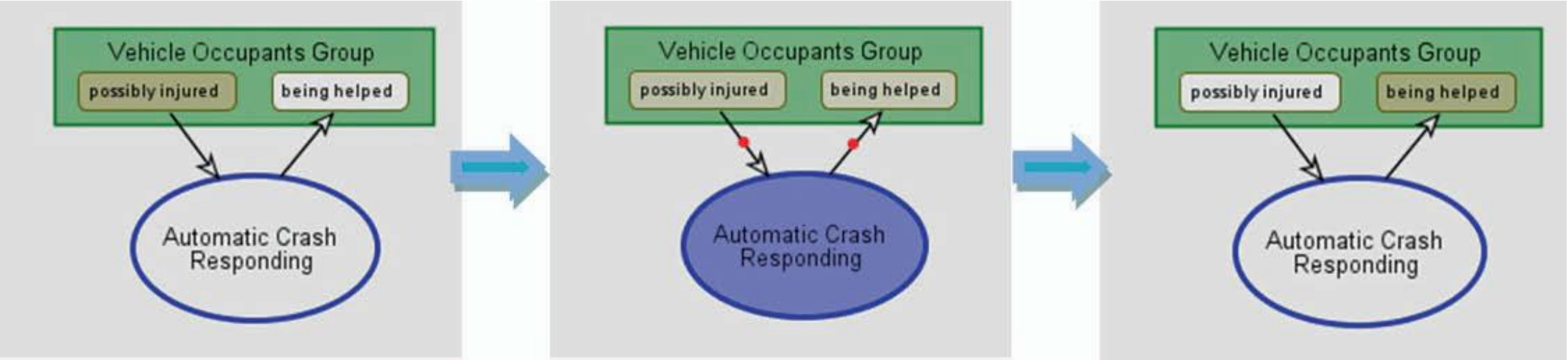
# Cognitive channels: visual-OPD and textual-OPL

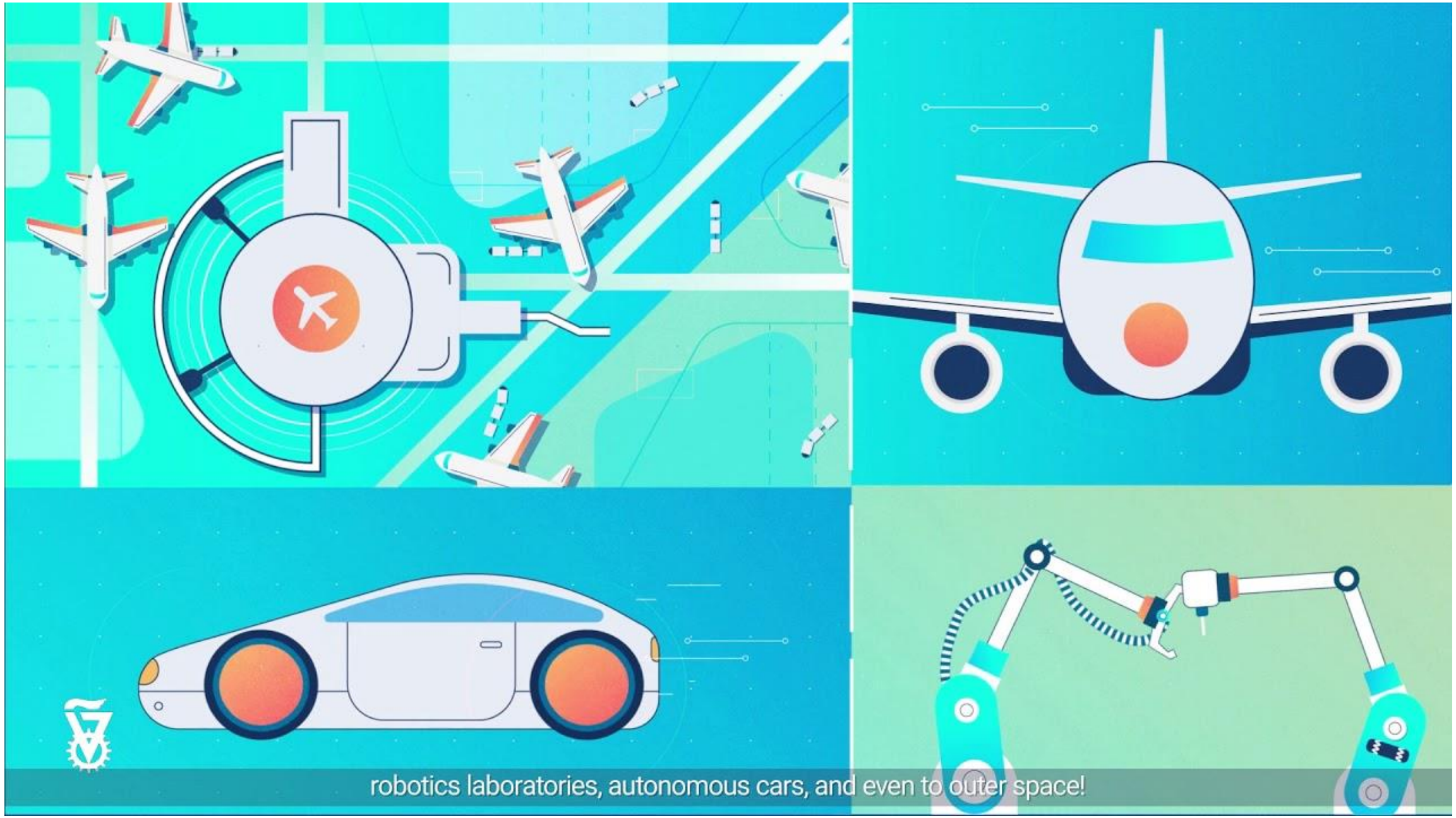



Christopher is physical.
Christopher knows OPM Theory.
Christopher handles presenting.
OPM consists of OPM Theory.
presenting is physical.
presenting consumes OPM Theory.
presenting yields Presentation.

# Model Simulation

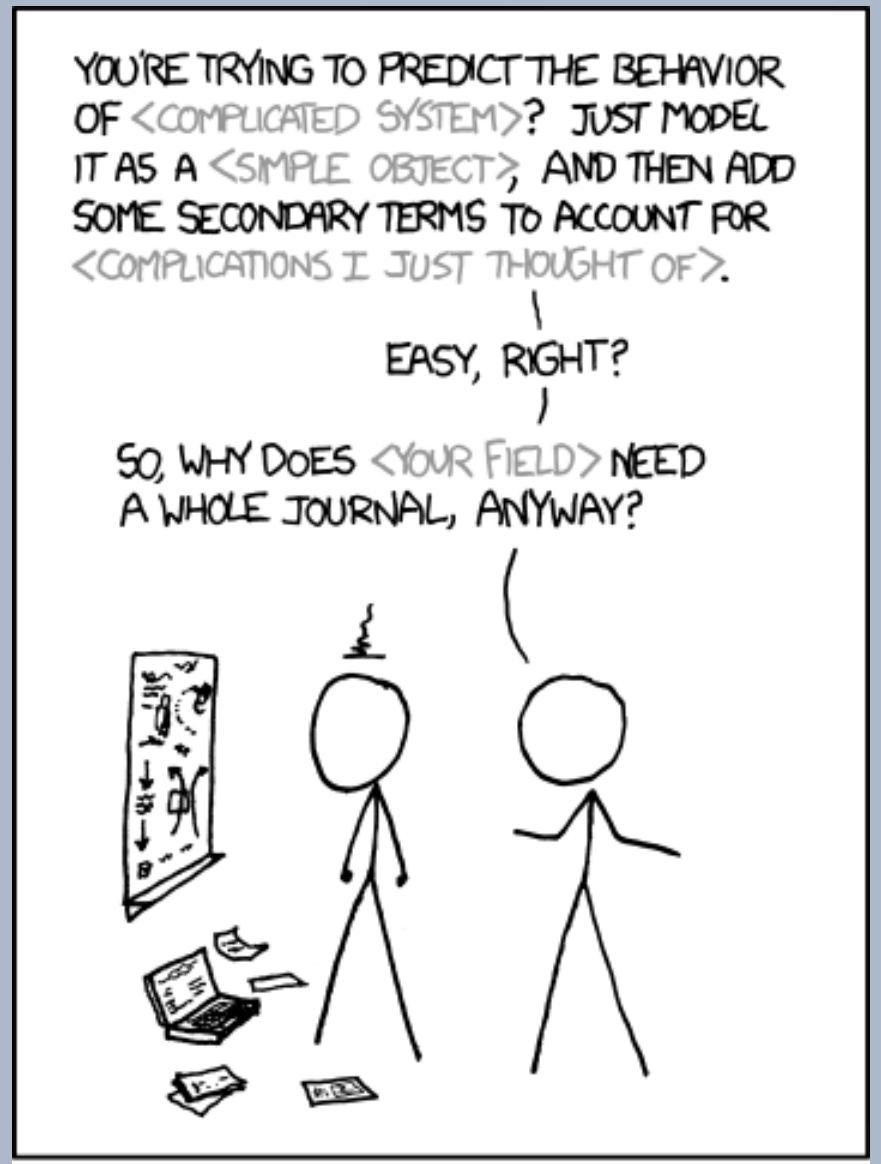
- One of the most attractive and useful features of an OPM model, which enables it to be visualized and tested, is its executability; that is, the ability to simulate a system by executing its model via animation in a properly designed software environment.






https://www.opcloud.tech/

# Building Blocks

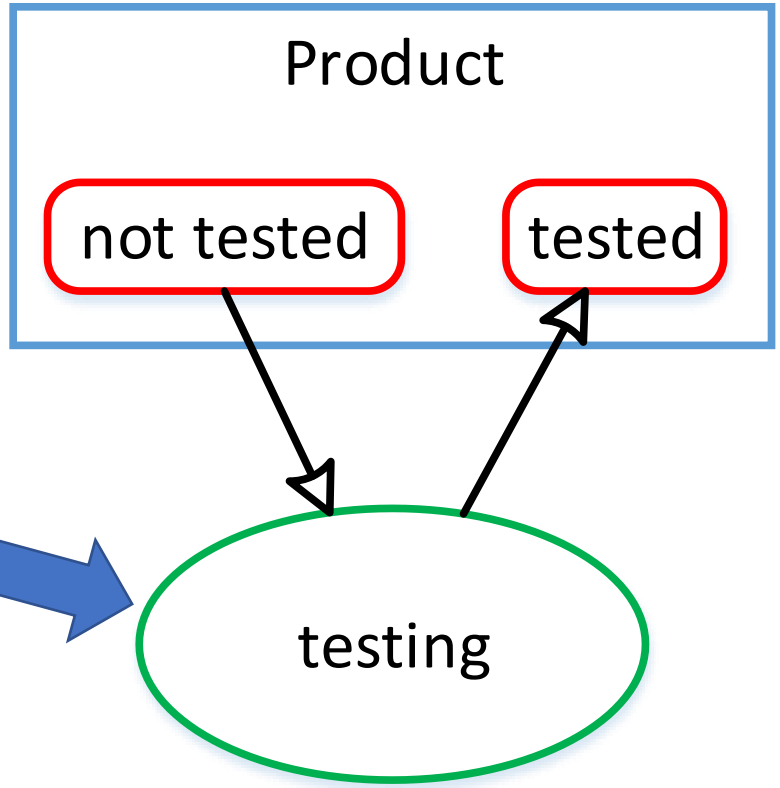



LIBERAL-ARTS MAJORS MAY BE ANNOYING SOMETIMES, BUT THERE'S *NOTHING* MORE OBNOXIOUS THAN A PHYSICIST FIRST ENCOUNTERING A NEW SUBJECT.

# OPM has only two building blocks:

1. **Objects with states**
   A thing that exists or might exist.

2. **Processes**
   A thing that transforms.



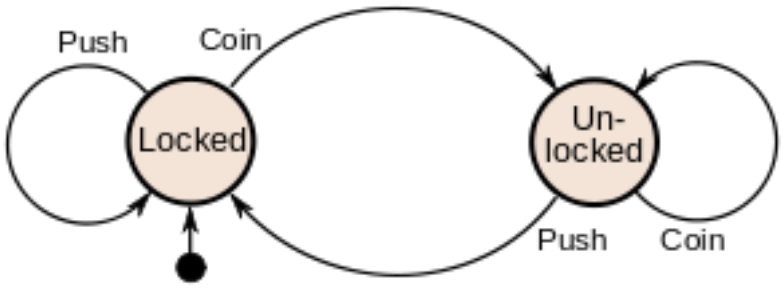

All the other elements are **relations** between things, expressed graphically as links

# A Object Thing

- An **object** is a **thing** that exists. Its existence can be either physical or conceptual. An **object** is a **thing** that can be transformed.
- An **object** can represent simple things such as car keys, or complex systems such as manufacturing plants.
- The graphical representation of an **object** in
- OPM is a square:

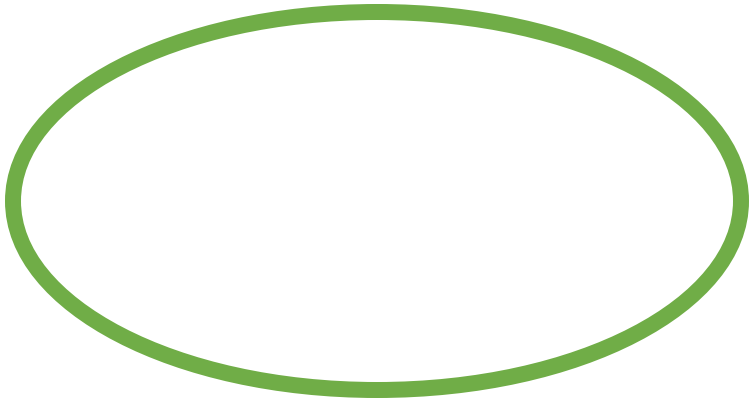
# States and Values

- A **state** is a possible situation at which an object can be.
- A **state** is only meaningful in within the context of a containing object.
- The graphical representation of a **state** in OPM is a rountangle (rounded rectangle):




# A Process Thing

- A **process** is a thing that transforms an **object**.
- This transformation can be:
  - Creation of an **object**.
  - Consumption of an **object**.
  - Changing the state of an **object**.
- By definition, a process must be associated with at least one object.
- The graphical representation of a process in OPM is an ellipse:



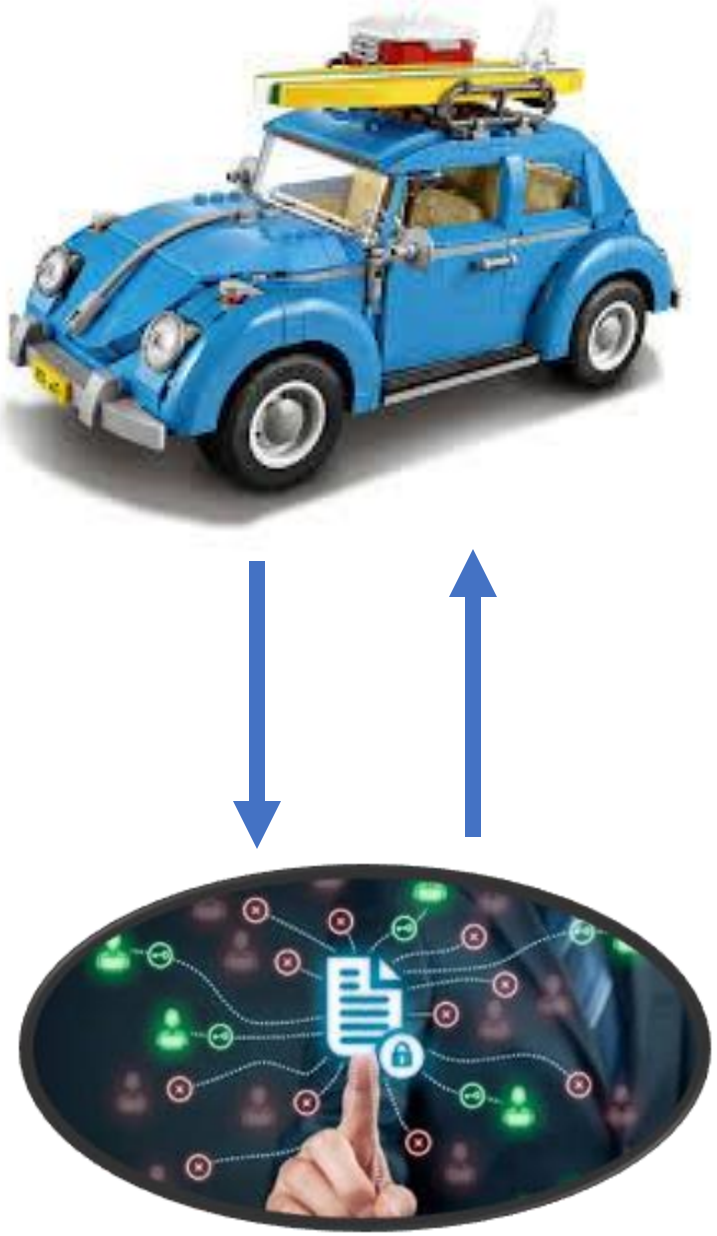
# Things nature

- Things
  - **Existential** essence
  - Design concern of the system or outside the **boundary**

# Essence

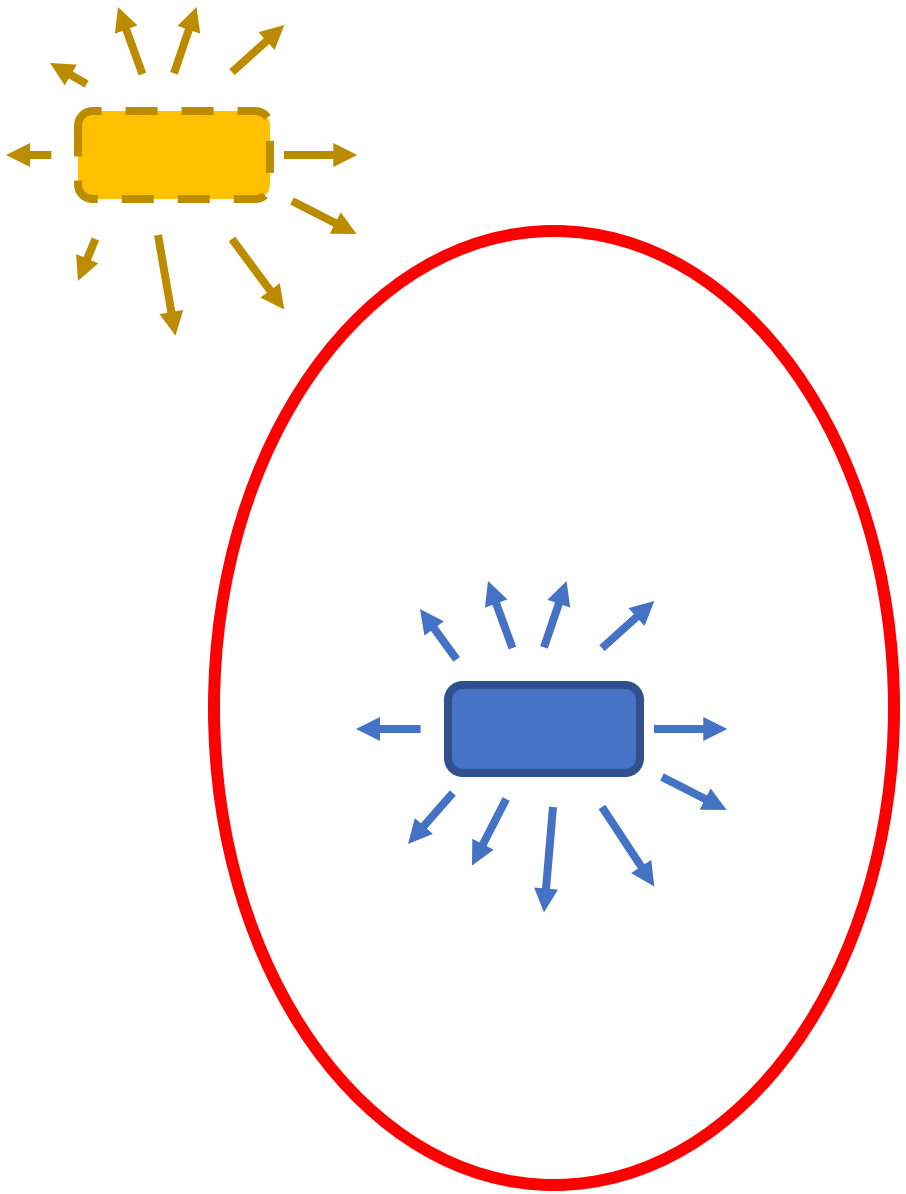
- **Key to modeling Cyber-Physical Systems**
  - **Physical** objects in the model represent what is really “out there” –actual states and values of objects
  - **Informatical** objects represent information about their corresponding physical objects

- Only ***informatical objects*** are available to a decision-making agent (human or artificial)

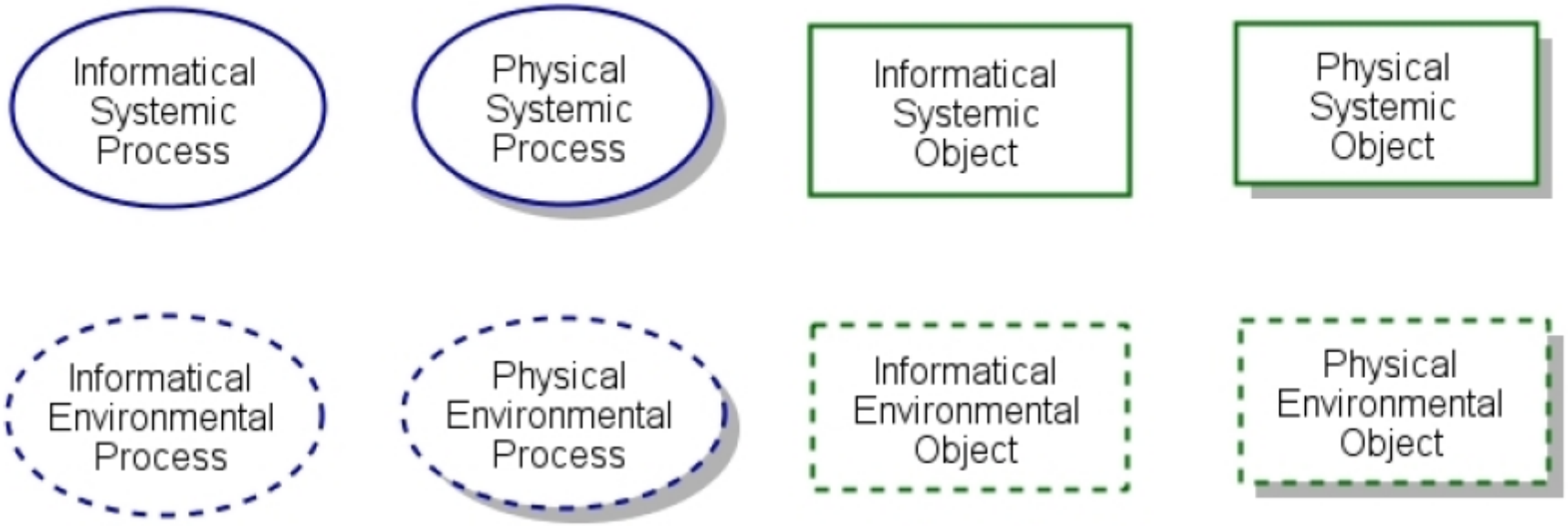


# Affiliation

- **Affiliation**, which pertains to the thing's scope and denotes whether the thing is:
  - **systemic**, i.e. part of the system, or
  - **environmental**, i.e. part of the system's environment.



# Essence/Affiliation




**Informatical Systemic Process** is an informatical and systemic process.
**Physical Systemic Process** is a physical and systemic process.
**Informatical Systemic Object** is an informatical and systemic object.
**Physical Systemic Object** is a physical and systemic object.
**Informatical Environmental Process** is an informatical and environmental process.
**Physical Environmental Process** is a physical and environmental process.
**Informatical Environmental Object** is an informatical and environmental object.
**Physical Environmental Object** is a physical and environmental object.

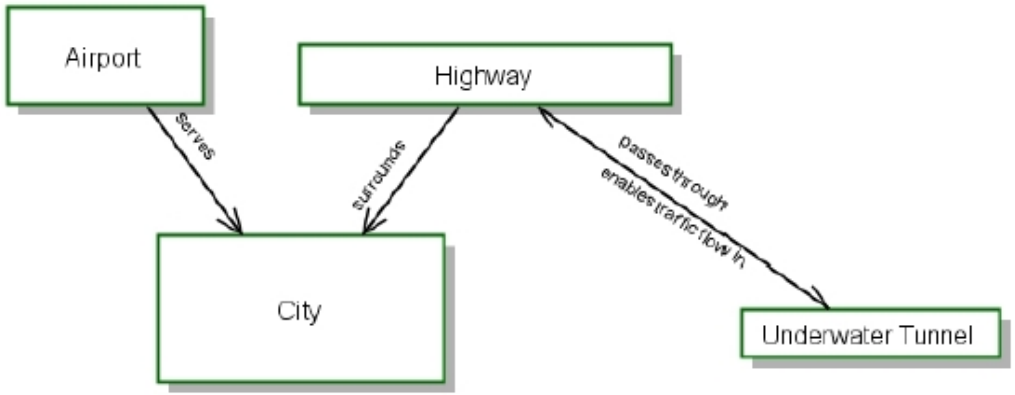
# OPM Structure

Structural Links

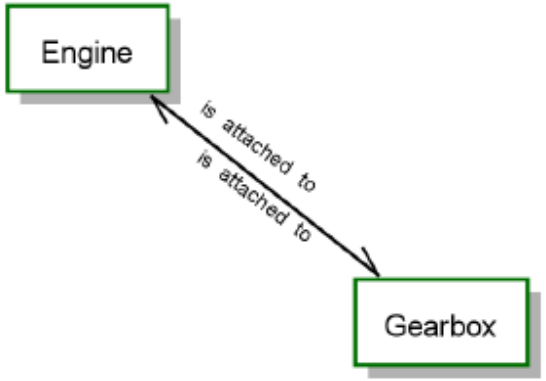
- **Structural Link** is a link that specificies **a static aspect** of the system by connecting na object to another object or a process to another process.

# Tagged Structural Links

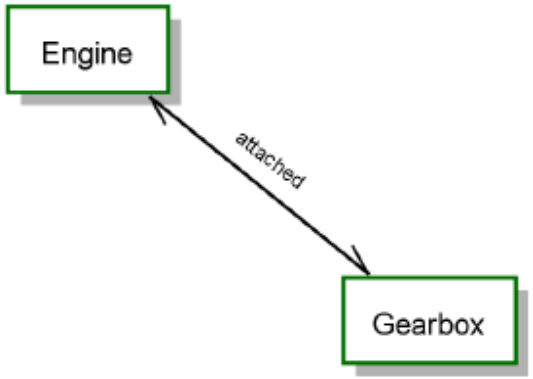
- A unidirectional tagged structural link defines a structural link between a source **object** and a **target** object.
  - The syntax of the unidirectional tagged structural link OPL sentence shall be: ***Source-thing tag Destination-thing***.
  - The syntax of the unidirectional null-tagged structural link OPL sentence shall be: ***Source-thing*** *relates to* ***Destination-thing***.
  - The syntax of the reciprocal tagged structural link with only one tag shall be: ***Source-thing*** *and* ***Destination-thing*** *are* ***reciprocity-tag***.
  - The syntax of the reciprocal tagged structural link with no tag shall be: ***Source-thing*** *and* ***Destination-thing*** *are related.*




Airport serves City.
Highway surrounds City.
Highway passes through Underwater Tunnel.
Underwater Tunnel enables traffic flow in Highway.



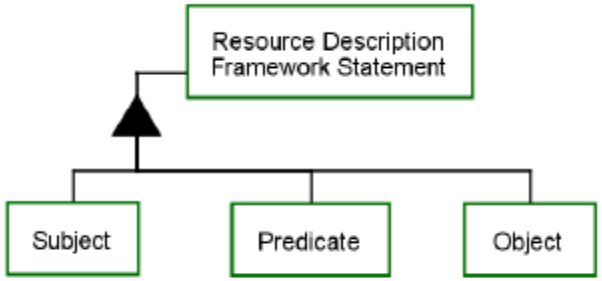

Engine is attached to Gearbox.
Gearbox is attached to Engine.



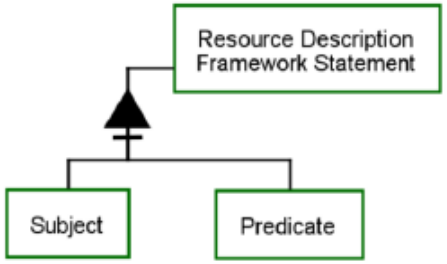

Engine and Gearbox are attached.

# Aggregation-Participation

- A structural relation which denotes that one (high level) **thing** aggregates (i.e. consists of, contains) one or more (low level) **things**. The high-level **thing** is called the *whole* or the *aggregate* while the lower-level **things** are called the *parts*.
  - The syntax of the aggregation-participation relation link shall be: ***Whole-thing*** *consists of* ***Part-thing1***, ***Part-thing2***, …, ***and Part-thingn***.




**Resource Description Framework Statement** consists of **Subject**, **Predicate**, and **Object**.




**Resource Description Framework Statement** consists of **Subject**, **Predicate**, and at least one other part.

# Exhibition-Characterization

- A structural relation which denotes that one **thing** exhibits (or is characterized) by another **thing**. A **thing** exhibits *features* that characterize it: An *attribute* is a *static feature* & An *operation* is a *dynamic feature*.
- The main difference between exhibition and aggregation is that an attribute always has a value, whether a part may be inexistent: A bag of candies will be empty after Purim (aggregation), but it will always have a color (exhibition).
  - The syntax of the exhibition-characterization relation link for an object exhibitor with a complete collection of n attributes and m operations shall be: ***Object-exhibitor exhibits Attribute1, Attribute2, … , and Attributen,*** *as well as* ***Operation1, Operator2, … , Operatorm.***
  - The syntax of the exhibition-characterization relation link for a process exhibitor with a complete collection of n operation features and m attribute features shall be: ***Process-exhibitor exhibits Operation1, Operator2, … , Operatorn,*** *as well as* ***Attribute1, Attribute2, …, and Attributem.***

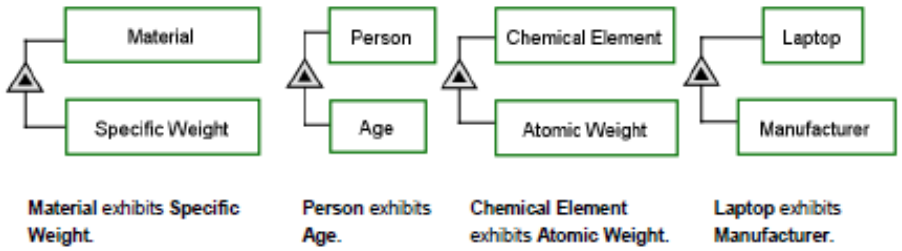



Figure 20 — Object attribute examples

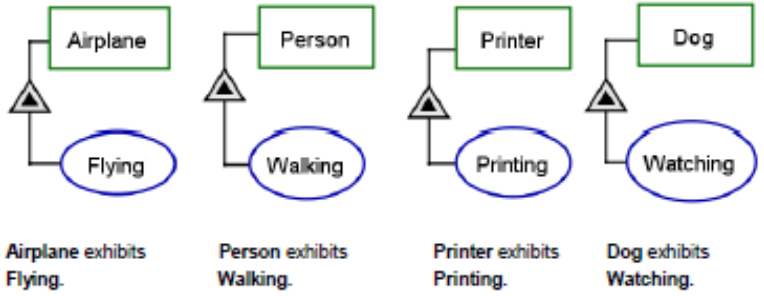



Figure 21 — Object exhibitor with operation examples

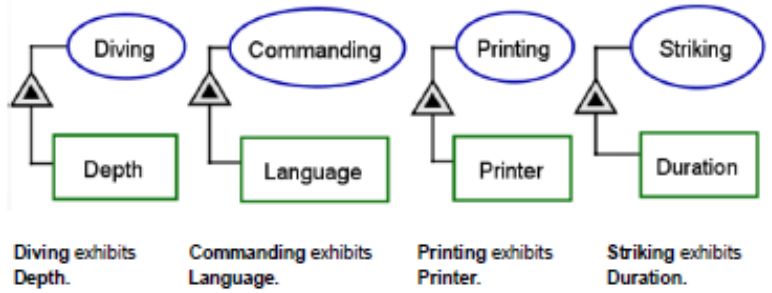
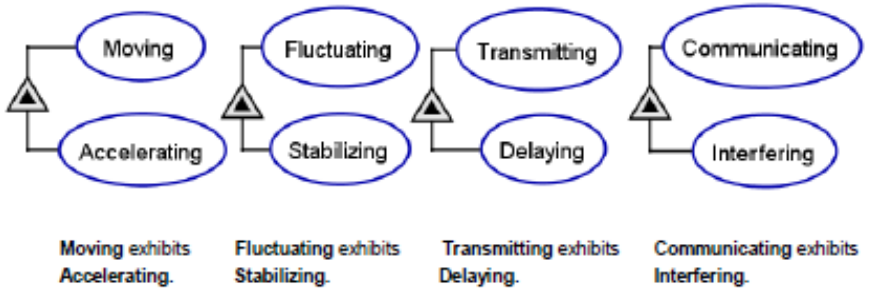
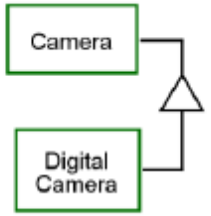



Figure 22 — Process exhibitor with attribute examples



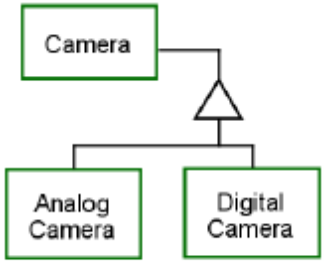
# Generalization-Specialization

- A structural relation which denotes that one **thing** specializes to another **thing**.
- Commonly referred as inheritance in OO modelling languages.
  - For a complete collection of n specializations of a general that is an object, the syntax of the generalization-specialization relation link OPL sentence shall be: ***Specialization-object1, Specialization-object2, …,*** *and* ***pecialization-objectn are General-object*****.**
  - For a complete collection of n specializations of a general that is a process, the syntax of the generalization-specialization relation link OPL sentence shall be: ***Specialization-process1, Specialization-process2, …,*** *and* ***Specialization-processn are General-process*****.**

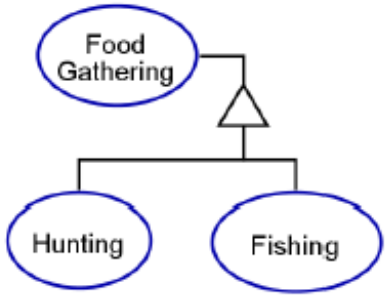



Digital Camera is a Camera.


Hunting is Food Gathering.



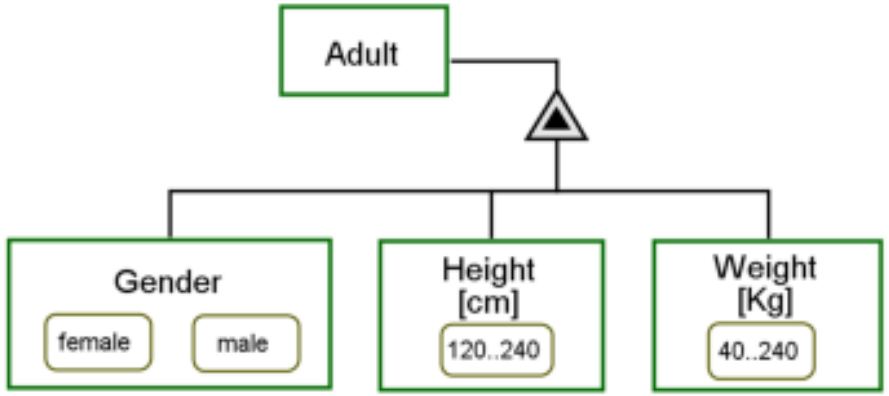

Analog Camera and Digital Camera are Cameras.



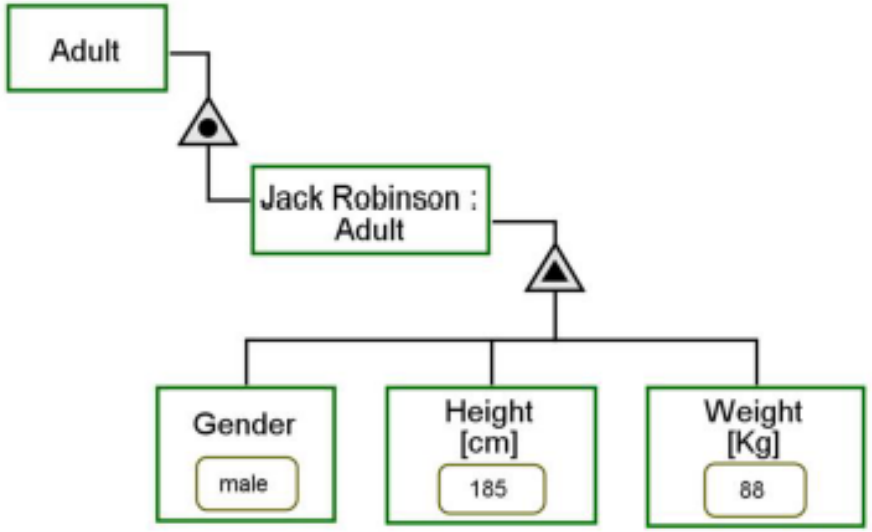

Hunting and Fishing are Food Gathering.

# Classification-Instantiation

- The classification, which is an object class or a process class, is a source pattern for a thing connecting with one or more destination things, which are instances of the source thing's pattern, i.e. the qualities the pattern specifies acquire explicit values to instantiate the instance thing.
- An instance of a class shall be an incarnation of a particular identifiable instance of that class with the same classification identifier.
  - The syntax of the classification-instantiation relation link between an object class and a single instance shall be: ***Instance-object is an instance of Class-object.***
  - The syntax of the classification-instantiation relation link between a process class and a single instance shall be: ***Instance-process is an instance of Class-process.***
  - The syntax of the classification-instantiation relation link between a process class and n instances shall be; ***Instance-object1, Instance-object2, …, Instance-objectn are instances of Class-object.***
  - The syntax of the classification-instantiation relation link between a process class and n instances shall be; ***Instance-process1, Instance-process2, …, Instance-processn are instances of Class-process.***

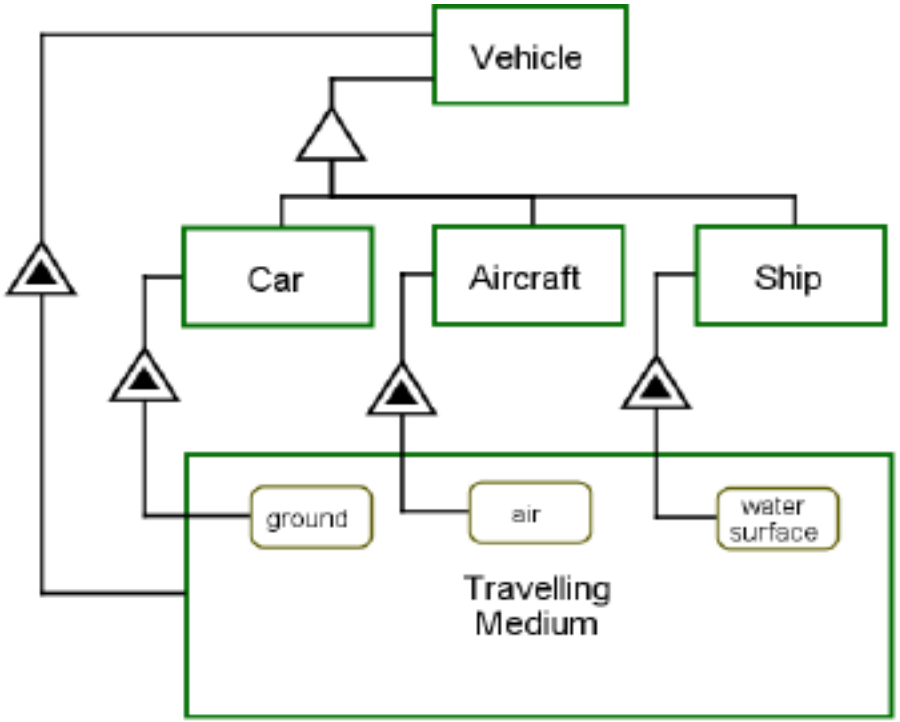



Adult exhibits Gender, Height in cm, and Weight in Kg.
Gender of Adult can be female or male.
Height in cm of Adult ranges from 120 to 240.
Weight in Kg of Adult range from 40 to 240.



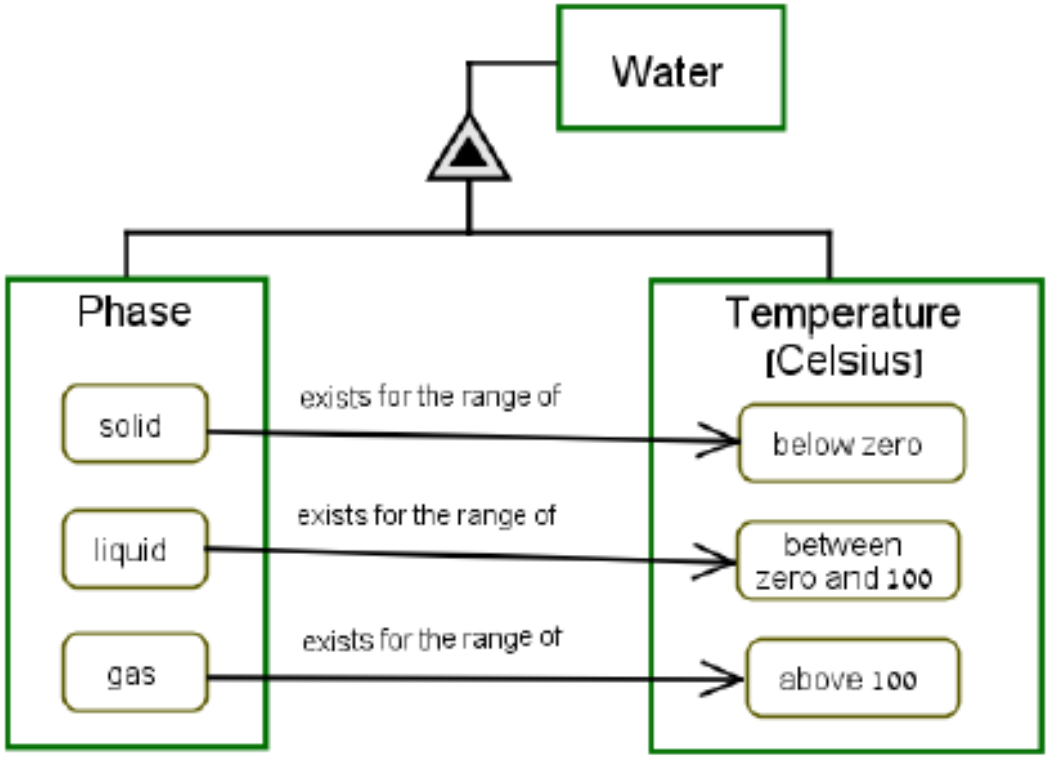

Jack Robinson is an instance of Adult.
Gender of Jack Robinson is male.
Height in cm of Jack Robinson is 185.
Weight in kg of Jack Robinson is 88.

# States can be also used in some relations



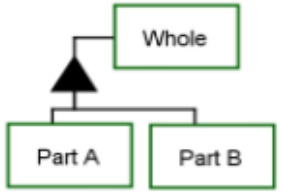
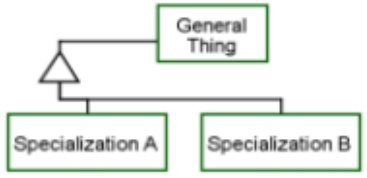
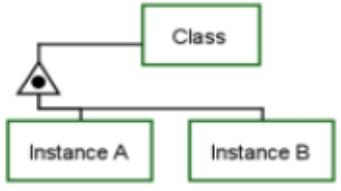
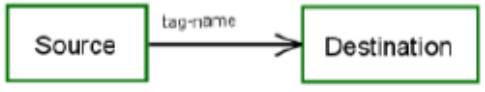
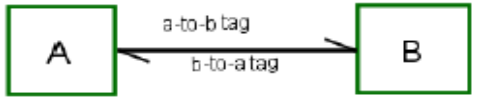
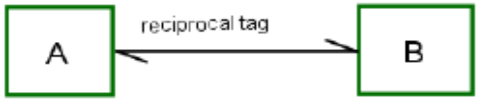

**Vehicle** exhibits **Travelling Medium.**
**Travelling Medium** of **Vehicle** can be **ground**, **air**, and **water surface.**
**Car**, **Aircraft**, and **Ship** are **Vehicles.**

**Car** exhibits **ground Travelling Medium.**
**Aircraft** exhibits **air Travelling Medium.**
**Ship** exhibits **water surface Travelling Medium.**




**Water** exhibits **Phase** and **Temperature** in **Celsius.**
**Phase** can be **solid**, **liquid**, or **gas.**
**Temperature** in **Celsius** can be **below zero**, **between zero and 100**, or **above 100.**
**Solid Phase exists for the range of below zero Temperature in Celsius.**
**Liquid Phase exists for the range of between zero and 100 Temperature in Celsius.**
**Gas Phase exists for the range of above 100 Temperature in Celsius.**

# Summary

| | | | |
|---|---|---|---|
| **Aggregation**-<br>Participation |  | **Whole** consists of **Part A** and **Part B**. | – |
| Exhibition-<br>**Characterization** |  | **Exhibitor** exhibits **Attribute A** as well as **Operation B**. | – |
| **Generalization**-<br>Specialization |  | – | **Specialization A** and **Specialization B** are **General Thing.** |
| **Classification**-<br>Instantiation |  | – | **Instance A** and **Instance B** are instances of **Class.** |
| Unidirectional tagged<br><br>[Unidirectional null tagged] |  | **Source tag-name Destination.**<br><br>[**Source** relates to **Destination.**] | |
| Bidirectional tagged |  | **A a-to-b tag B.**<br>**B b-to-a tag A.** | |
| Reciprocal tagged<br><br>[Reciprocal null tagged] |  | **A** and **B** are **reciprocal tag.**<br><br>[**A** and **B** are related.] | |

# OPM Behavior

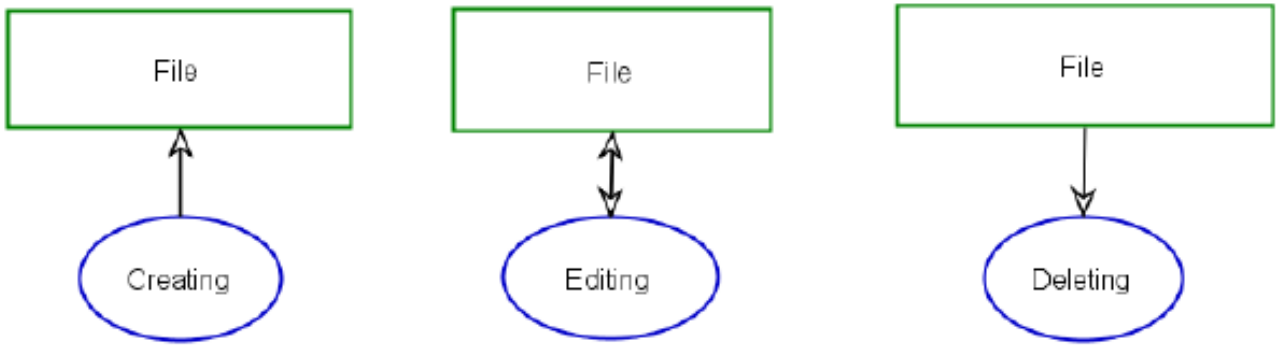
Procedural Links

- **Procedural Link** is a link that specifies a **dynamic aspect** of the system by connecting na object (or one of its states) and a process.

- Procedural links symbolize the behavior of the modeled system.
- Three types:
  - Enabling links: link a **process** to an **object** that enables the **process** but is not affected by it.
  - Transforming links: links a **process** to an **object** that is affected by the **process**.
  - Invocation links: shortcut notation between two consecutive **processes**.

# Transforming Links

- A transformation link is a procedural link that connects a process with an object transformed by the process. Three types:
  - **Consumption**: the linked object is consumed and eliminated by the process.
  - **Result**: the linked object is constructed by the process.
  - **Effect**: the linked object is changed by the process.




Creating yields File | Editing affects File | Deleting consumes File

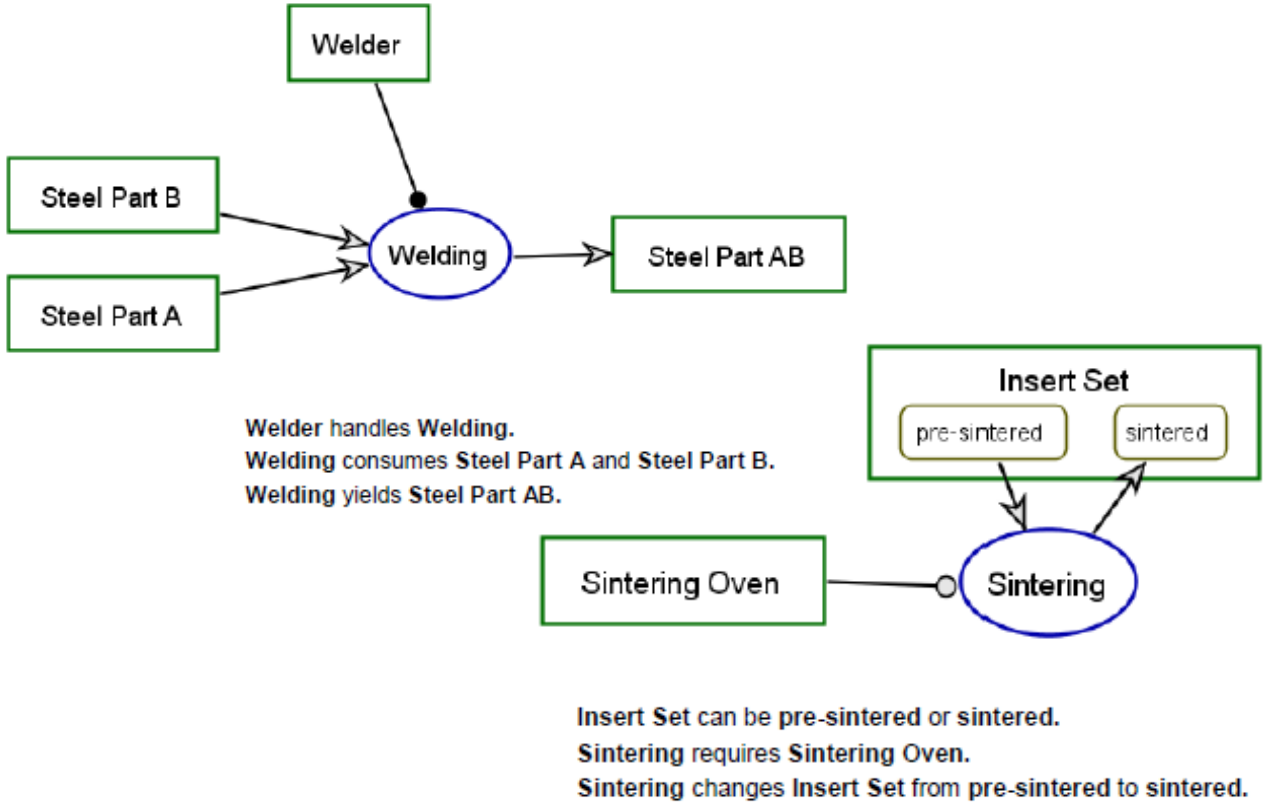
The syntax of a consumption link OPL sentence shall be: ***Processing consumes Consumee*.**

The syntax of a result link OPL sentence shall be: ***Processing yields Resultee*.**

The syntax of an effect link OPL sentence shall be: ***Processing affects Affectee*.**

# Enabling Links

- An enabling link is a procedural link that connects a process with an enabler object of that process. Two types:
  - **Agent**: an enabler who is a human or a group of humans.
  - **Instrument**: a non-human enabler

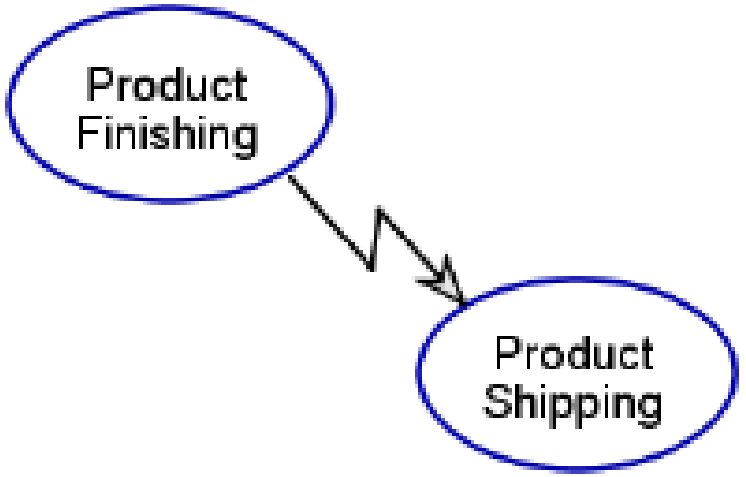



The syntax of an agent link OPL sentence shall be: ***Agent handles Processing*.**

The syntax of an instrument link OPL sentence shall be: ***Processing requires Instrument*.**

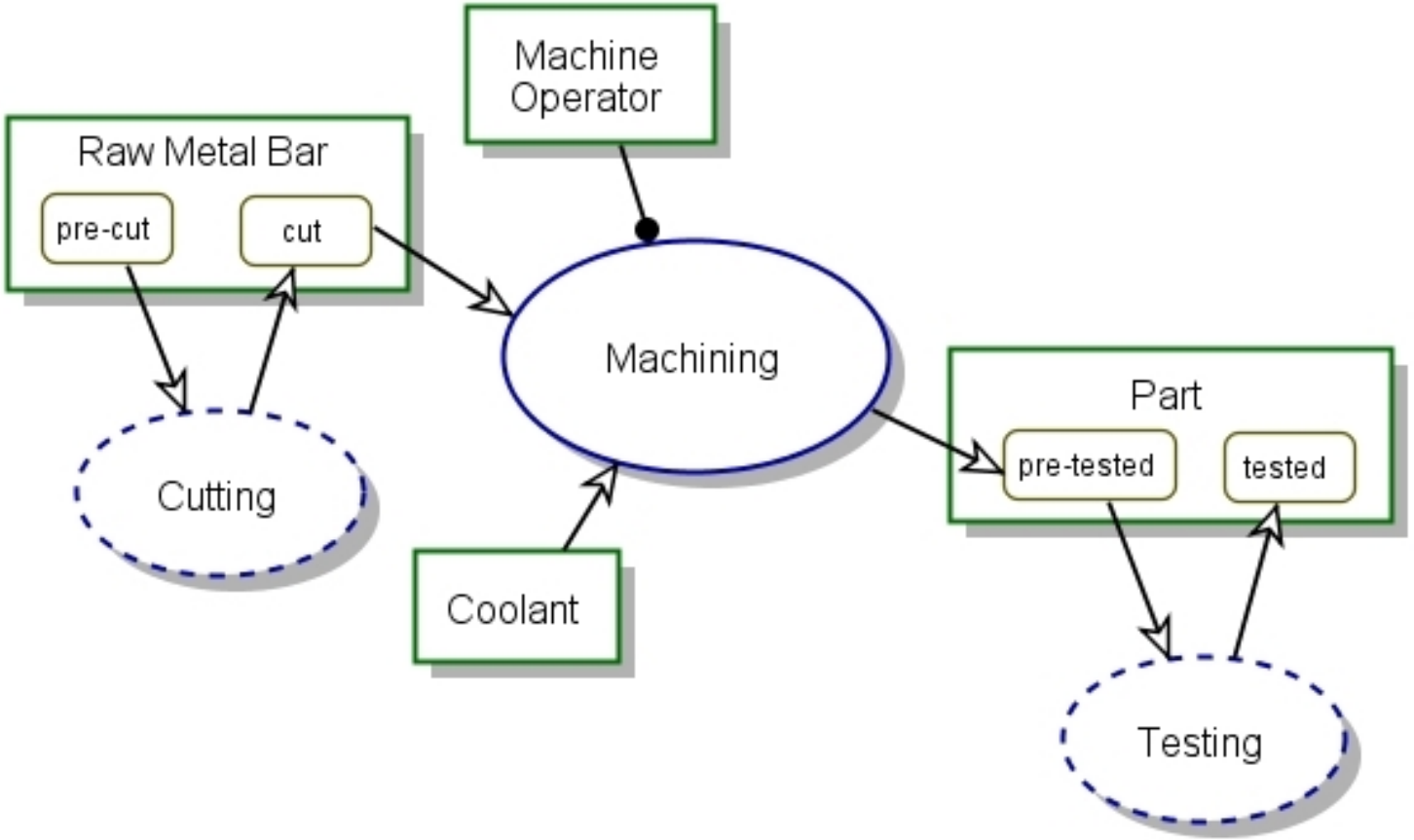
# Invocation Link

- By definition, a process must transform an object. But sometimes the result of a process is not significant to the system and may be ignored. However, the result of the process is significant to a consecutive process.
- The invocation link provides a shortcut to bypass the modeling of the irrelevant object.
  - The syntax of an invocation link OPL sentence shall be: ***Invoking-process invokes invoked-process***.
  - The syntax of a self-invocation link OPL sentence shall be: ***Invoking-process invokes itself***.

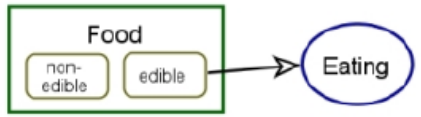
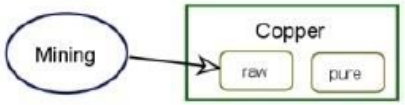
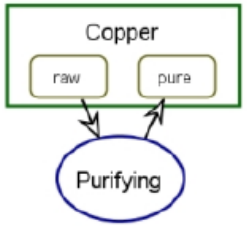
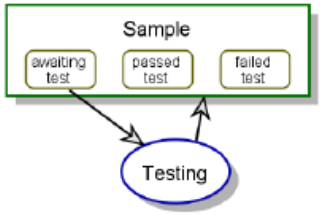


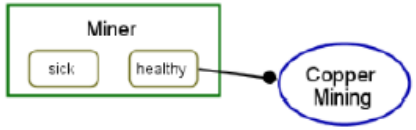
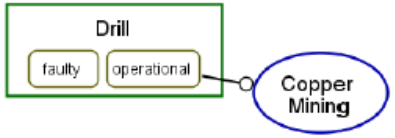

**Product Finishing** invokes **Product Shipping**.

# States can be also used in some relations



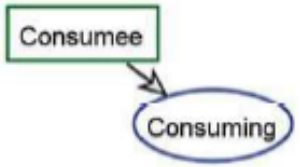
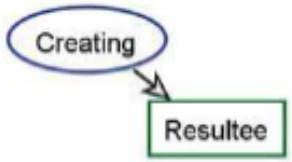
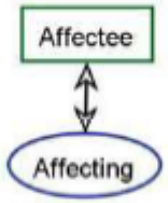
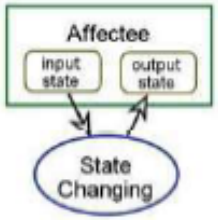

Raw Metal Bar is physical.
Raw Metal Bar can be pre-cut or cut.
Machine Operator is physical.
Coolant is physical.
Machining is physical.
Machining requires Coolant.
Machine Operator handles Machining.
Part is physical.
Part can be pre-tested or tested.
Testing is environmental and physical.
Cutting changes Raw Metal Bar from pre-cut to cut.
Machining consumes Raw Metal Bar.
Machining yields pre-tested Part.
Testing changes Part from pre-tested to tested.

| Name | Semantics | Sample OPD & OPL | Source | Destination |
|---|---|---|---|---|
| **State-specified consumption link** | The process consumes the object if and only if the object is in the specified state. |  <br>**Eating** consumes **edible Food.** | consumee state | process |
| **State-specified result link** | The process generates the object in the specified state. |  <br>**Mining** yields **raw Copper.** | process | resultee state |
| **Input-output-specified effect link pair** (consisting of one state-specified *input link* and one state-specified *output link*) | The process changes the object from a specified input state via the *input link* to a specified output state via the *output link*. |  <br>**Purifying** changes **Copper** from **raw** to **pure.** | affectee source state<br>affecting process | affecting process<br>affectee destination state |
| **Input-specified effect link pair** (consisting of one state-specified *input link* and one state-unspecified *output link*) | The process changes the object from a specified input state to any output state. |  <br>**Testing** changes **Sample** from **awaiting test.** | affectee source state<br>affecting process | affecting process<br>affectee |
| **Output-specified effect link pair** (consisting of one state-unspecified *input link* and one state-specified *output link*) | The process changes the object from any input state to a specified output state. |  <br>**Cleaning & Painting** changes **Engine Hood** to **painted.** | affectee<br>affecting process | affecting process<br>affectee destination state |

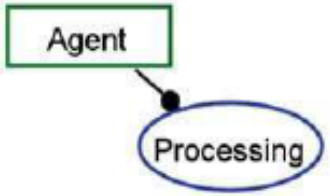
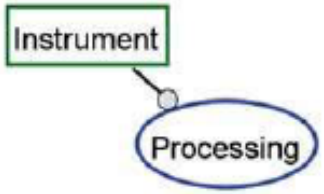
| Name | Semantics | Sample OPD & OPL | Source | Destination |
|---|---|---|---|---|
| **State-specified agent link** | The human agent enables the process provided she is at the specified state. |  <br>**Healthy Miner** handles **Copper Mining.** | agent state | enabled process |
| **State-specified instrument link** | The process requires the instrument at the specified state. |  <br>**Copper Mining** requires **operational Drill.** | instrument state | enabled process |

# Summary

Procedural transforming links

| consumption link | result link | effect link | in-out link pair |
|---|---|---|---|
|   |   |   |   |
| **Consuming** consumes **Consumee.** | **Creating** yields **Resultee.** | **Affecting** affects **Affectee.** | **State Changing** changes **Affectee** from **input state** to **output state.** |

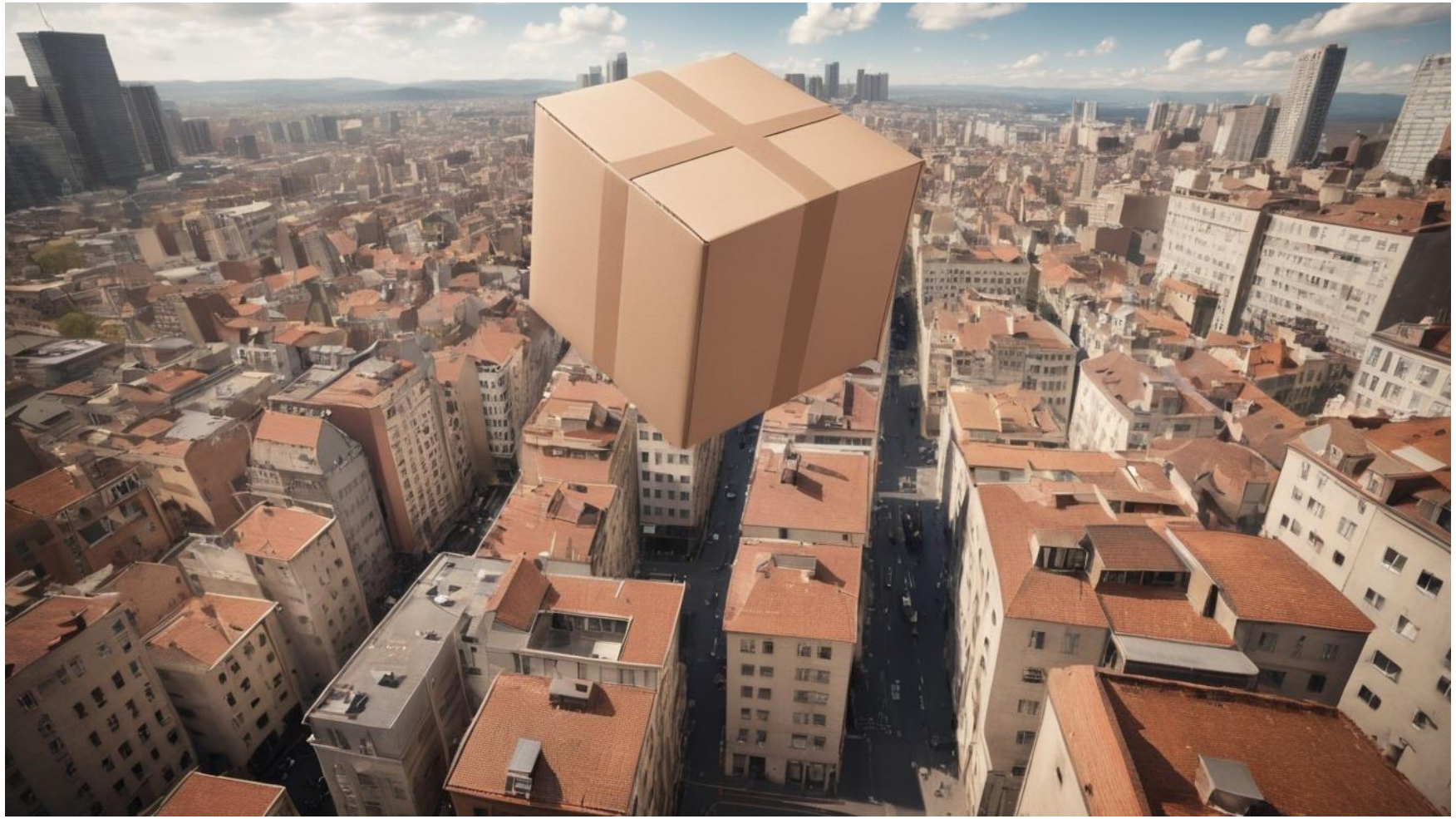
Procedural enabling links

| agent link | instrument link |
|---|---|
|   |   |
| **Agent** handles **Processing.** | **Processing** requires **Instrument.** |

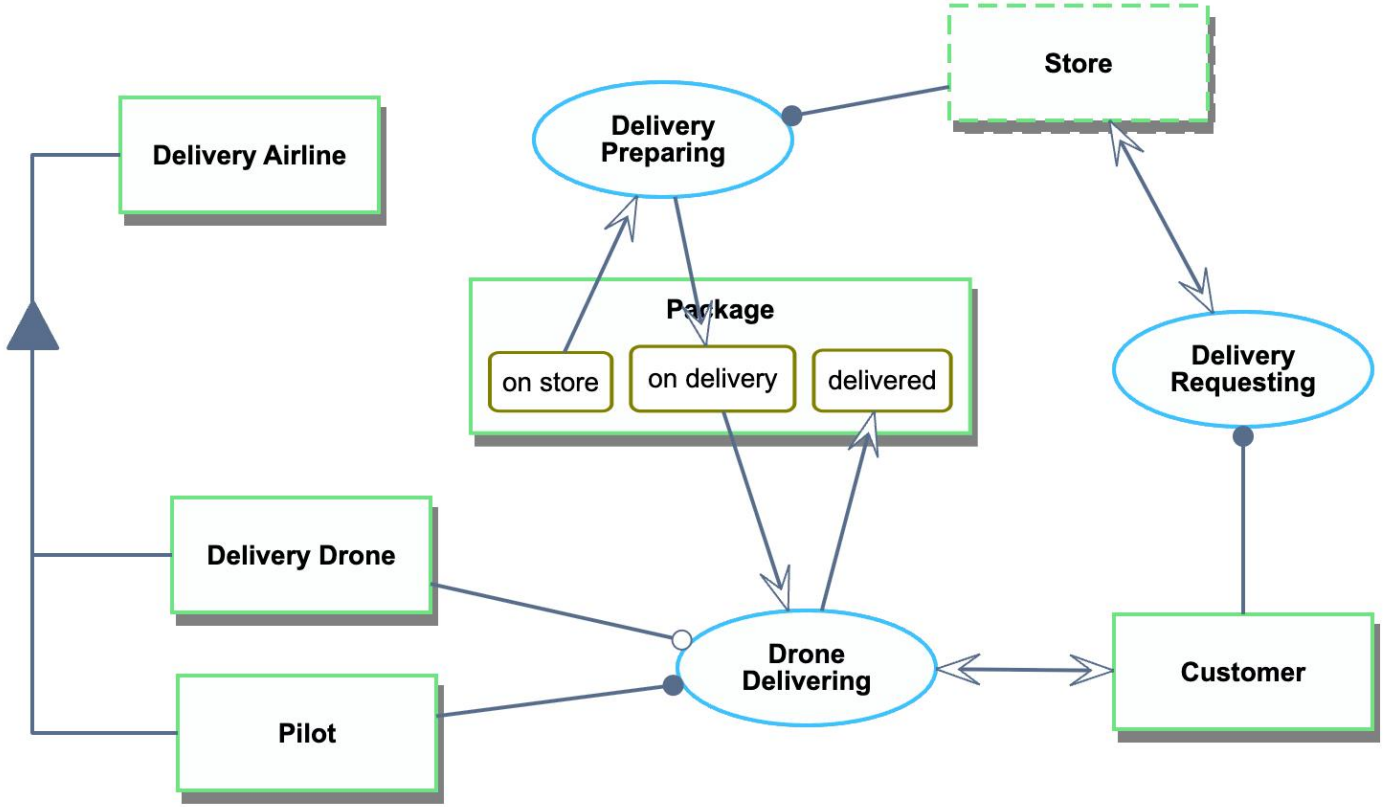
# One-page Simple Example

# Walking through one example

- **Aerial package delivery**



# Walking through one single page example




1. **Delivery Airline** is a physical object**.**
2. **Delivery Drone** is a physical object**.**
3. **Pilot** is a physical object**.**
4. **Package** is a physical object**.**
5. **Package** can be **delivered,on delivery** or **on store.**
6. **Customer** is a physical object**.**
7. **Store** is a physical and environmental object**.**
8. **Delivery Airline** consists of **Delivery Drone** and **Pilot.**
9. **Drone Delivering** changes **Package** from **on delivery** to **delivered.**
10. **Pilot** handles **Drone Delivering.**
11. **Drone Delivering** requires **Delivery Drone.**
12. **Drone Delivering** affects **Customer.**
13. **Delivery Preparing** changes **Package** from **on store** to **on delivery.**
14. **Store** handles **Delivery Preparing.**
15. **Customer** handles **Delivery Requesting.**
16. **Delivery Requesting** affects **Store.**

# 12 OPM Principles

- **1. The Function-as-a-Seed** – Modelling a system starts by defining, naming, and depicting the function of the system, which is also its top-level process.

- **2. The Model Fact Representation** – An OPM model fact needs to appear in at least one OPD in order for it to be represented in the model.

- **3. The Timeline** – The timeline wihin an in-zoomed process is directed by default from the top of the in-zoomed process elipse to its bottom

- **4. The Minimal Conceptual Modelling Language** – A symbol system – a language – that can conceptually model a given system using ontology with fewer diagram kinds and fewer symbosl and relations among them is prefeable over a larger ontology with mode diagram kinds and more symbols and relations among them.

- **5. The Thing Importance** – The importance of a thing T in an OPM Model is directly related to the highest OPD in the OPD hierarchy where T appears.

- **6. The Object Transformation by Process** – In a complete OPM Model, each process must be connected to at least one object that the process transforms or one state of the object that the process transforms.

- **7. The Procedural Link Uniqueness** – At any level of detail, na object and a process can be connected with at most one procedural link, which uniquely determines the role of the object with respect to the process.

- **8. The Singular Name** – A name of an OPM thing must be singular. Plural has to be converted to singular by adding the world “Set” for inanimate things or “Group” for humans.

- **9. The Graphics-Text Equivalence** – Any model fact expressed graphically in an OPD is also expressed textually in the corresponding OPL paragraph.

- **10. The Thing Name Uniqueness** – Different things in na OPM Model which are not features must have different names. Features are distinguishable by appending to them the reserved word “of” and the name of their exhibitor.

- **11. The Detail Hierarchy** – Whenever na OPD becames hard to comprehend due to an excessive amount of details, a new, descendant OPD shall be created.

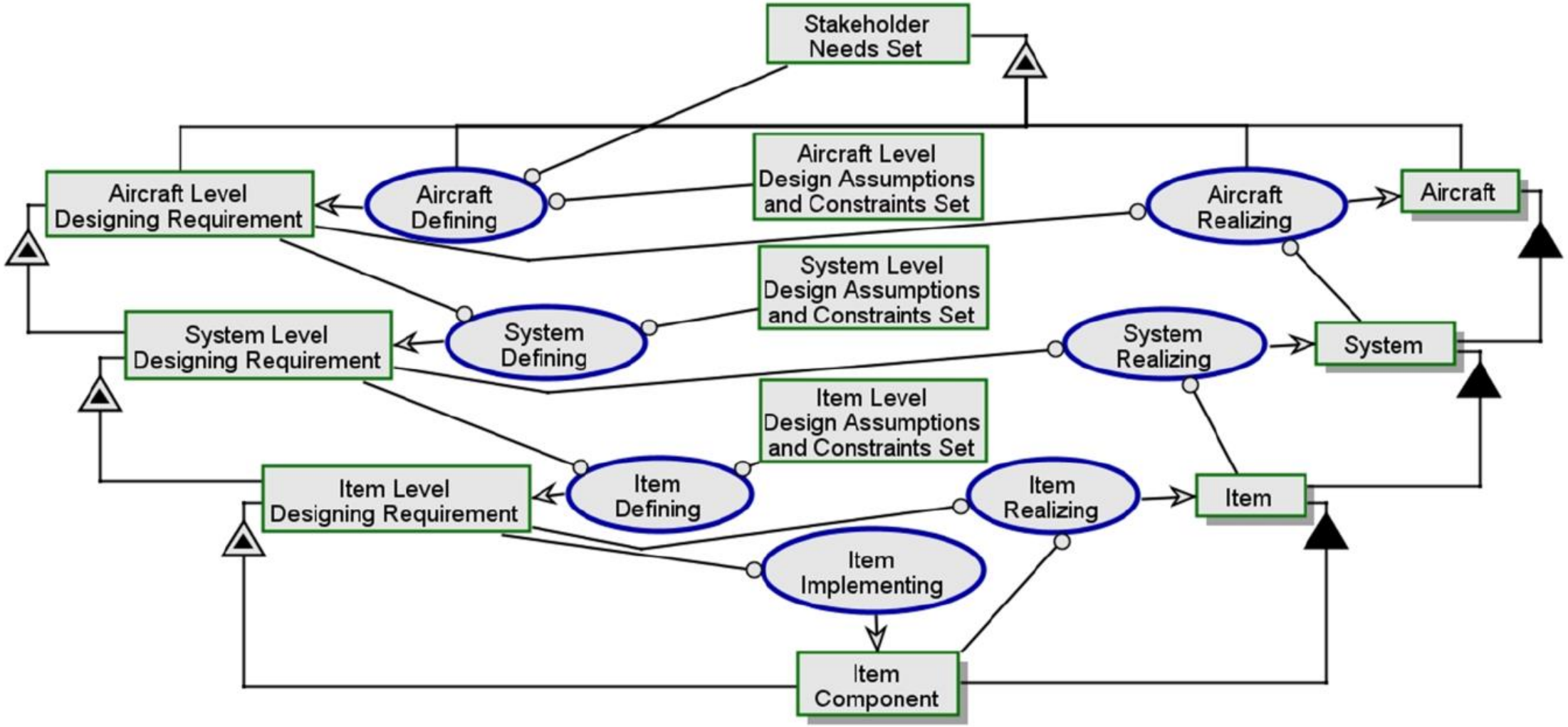
- **12. The Skip Semantics Precedence** – Skip semantics takes precedence over wait semantics.

# Final Remarks

# Model-Based Systems Engineering for Aircraft Design with Dynamic Landing Constraints Using Object-Process Methodology

Linwen Li[1 * † ‡], Natali Levi Soskin[2 †], Ahmad Jbara[3 †], Moti Karpel[4 ‡] and Dov Dori[5 †]

[*]*Shanghai Aircraft Design and Research Institute, Shanghai, 201210, China*




SOME EXAMPLES

# Some thoughts

- OPM is **simple and powerful to talk with stakeholders** and create the first architectures
- OPM uses **one diagram type** to handle structure and behavior
  - the language vocabulary has only a **couple of symbols** and **semantics to mimic common sketching**.
- OPM allows simple-formal modelling and enables to **control the complexity**.
- OPM is the only MBSE that **simulates CONCEPTS**.
- As it is an ISO, it is **worth a try**.
- OPM **lacks transformational tools** to other domains and a open metamodel (EMF).

# OPM is growing

- OPM main tool is the OPCloud (“web” based)
- It has been highly improved from the OPCat used through the course.
  - Usability is better
  - Allows dynamic behavior
  - IoT connectable through MQTT
  - Socket connection
  - Stereotypes
  - Styling
  - Timing