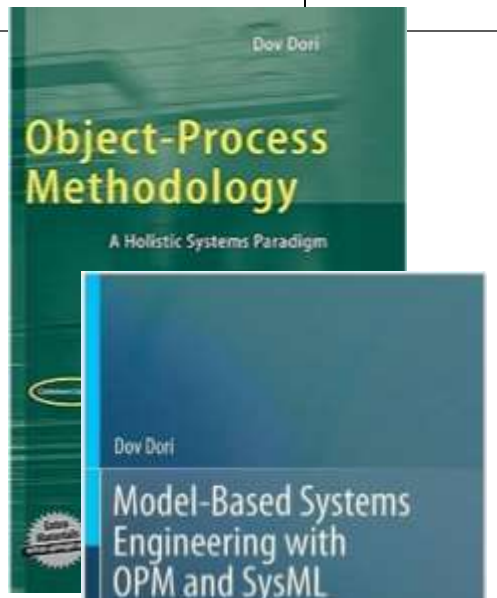
Technion
Israel Institute of Technology

ESM L
Enterprise Systems Modeling Laboratory

Massachusetts Institute of Technology

Object-Process Methodology: What is it, What is it Good For?

Dov Dori
Technion, Israel Institute of Technology
Massachusetts Institute of Technology



Applied Math Graduate Program
Technion
December 15, 2019

OCCAM's RAZOR

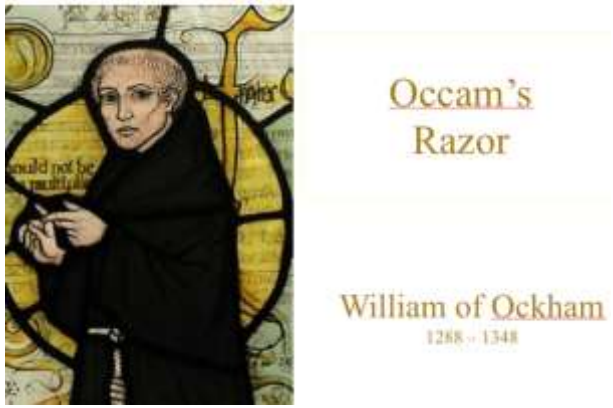
14th Century logician and Franciscan priest William of Ockham

"Entities should not be multiplied unnecessarily."

"Entia non sunt multiplicanda praeter necessitatem"

In an extended version:

"If you have two equally likely solutions to a problem, choose the simplest."



OCCAM's RAZOR is an important guiding principle of OPM

Theoretical Foundation 1: Three Cognitive Assumptions

(1) ***Dual-channel processing*** – humans possess separate systems for processing visual and verbal representations (Clark & Paivio, 1991; Baddeley, 1992).

(2) ***Limited capacity*** – the amount of processing that can take place within each information processing channel is extremely limited (Miller, 1956; Chandler & Sweller, 1991; Baddeley, 1992).

(3) ***Active processing*** – meaningful learning occurs during active cognitive processing, paying attention to words and pictures.

Mayer, R.E. (2003). The promise of multimedia learning: using the same instructional design methods across different media. Learning and Instruction, 13, pp. 125-139.

OPM Design Principle 1: Simplicity

- Simplicity is a must for modeling systems
- We cannot ignore the **inherent complexity** of systems.

However,

- We can simplify the way systems are modeled
  - without sacrificing accuracy, and
  - without sparing details.

OPM Design Principle 2: Minimum Description Length

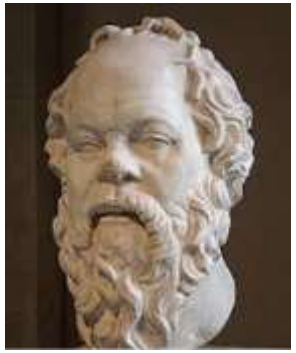
- If the same system can be expresses by two languages with the same level of fidelity, the one with the shortest description is preferred.
  - Inspired by The Minimum Description Length (MDL) Principle (Rissanen 1978)

Rissanen, J. (1978). Modeling by shortest data description. Automatica, 14: 465-471.

OPM Theoretical Foundation

Universal Ontology

Let's get to it via asking Socratic Questions



Theoretical Foundation 2: Ontology & Universal Ontology

Ontology:

a set of **concepts** and **relations** for describing a ***domain*** and systems within it.

Universal Ontology:

a ***domain-independent*** set of concepts and relations for describing systems in the universe, both natural and man-made.

Fundamental question 1:

What is needed to describe the universe and systems in it?

Answer:

Things and relations among them.

Question 2:

What is a thing
or
what can it do?

Answer:

A things can either
exist at some point in **time**
or
happen over **time**

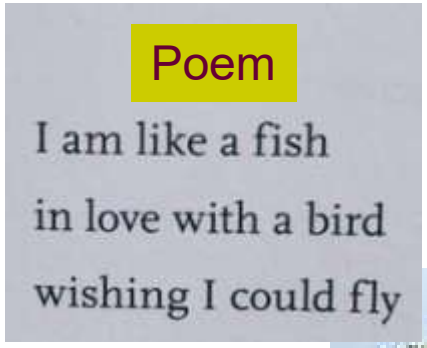
Thing 1 and **Thing 2** ***exist***.

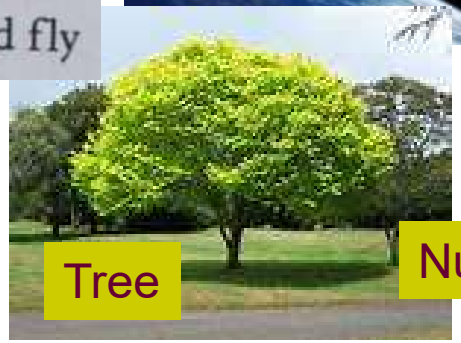
Running of Thing 1 and Thing 2 ***happens***.

Any **thing** can either **exist** or **happen** – nothing else;
Every **thing** can be classified into one of these two!

Question 3:

What are the things that *exist* in the world?





Answer:

Objects *exist.*

*They are **static** –*

time independent. *(syntactically: **nouns**)*

Question 4:

What are the things that *happen* in the world?

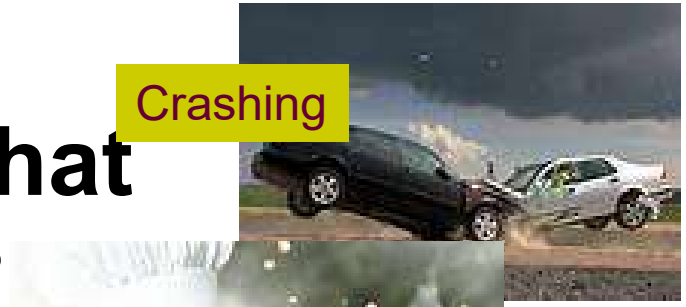
Answer:

Processes *happen.*

*They are **dynamic** –*

time dependent.

(syntactically:
verbs in gerund form: "…ing")



Processes do not happen in vacuum!

Question 5:

Whom do processes happen to?

Answer:

Processes happen to objects.

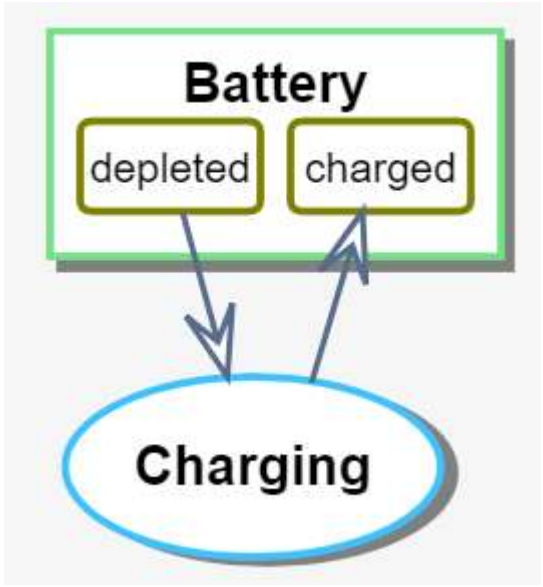
Question 6:

What does a process do when it happens to an object?

Answer:

The process *transforms* the Object.

Only two OPM Things: Objects and Processes



Charging changes Battery from depleted to charged.

Object: A thing that exists or might exist physically or informatically.

Process: A thing that transforms or might transform one or more objects.

processes *transform* objects.

Question 7:

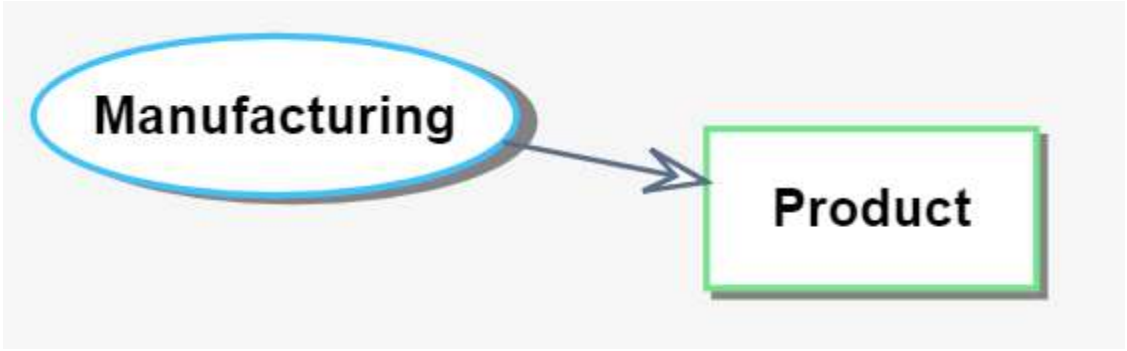
In what three ways does a process *transform* an object?

Answer:

1. *creating* an object
2. *destroying* an object
3. *affecting* an object.

The first kind of transforming: Generation

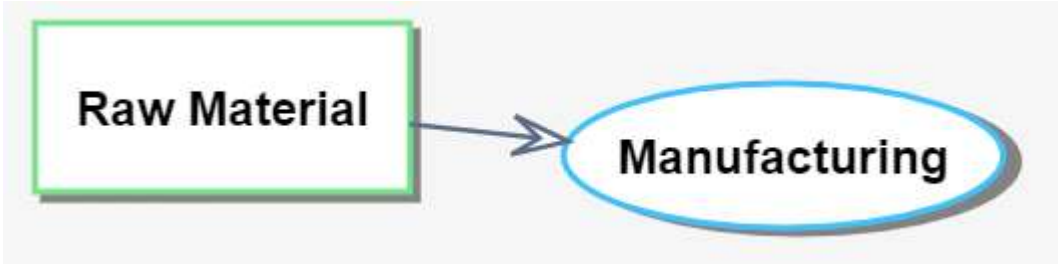
The process can ***create*** the object



Manufacturing yields Product.

The second kind of transforming: Consumption

The process can ***consume*** the object



Manufacturing consumes Raw Material.

The third kind of transforming: Effect

The process can ***affect*** the object

Question 8:

How does a process *affect* an object?

Answer:

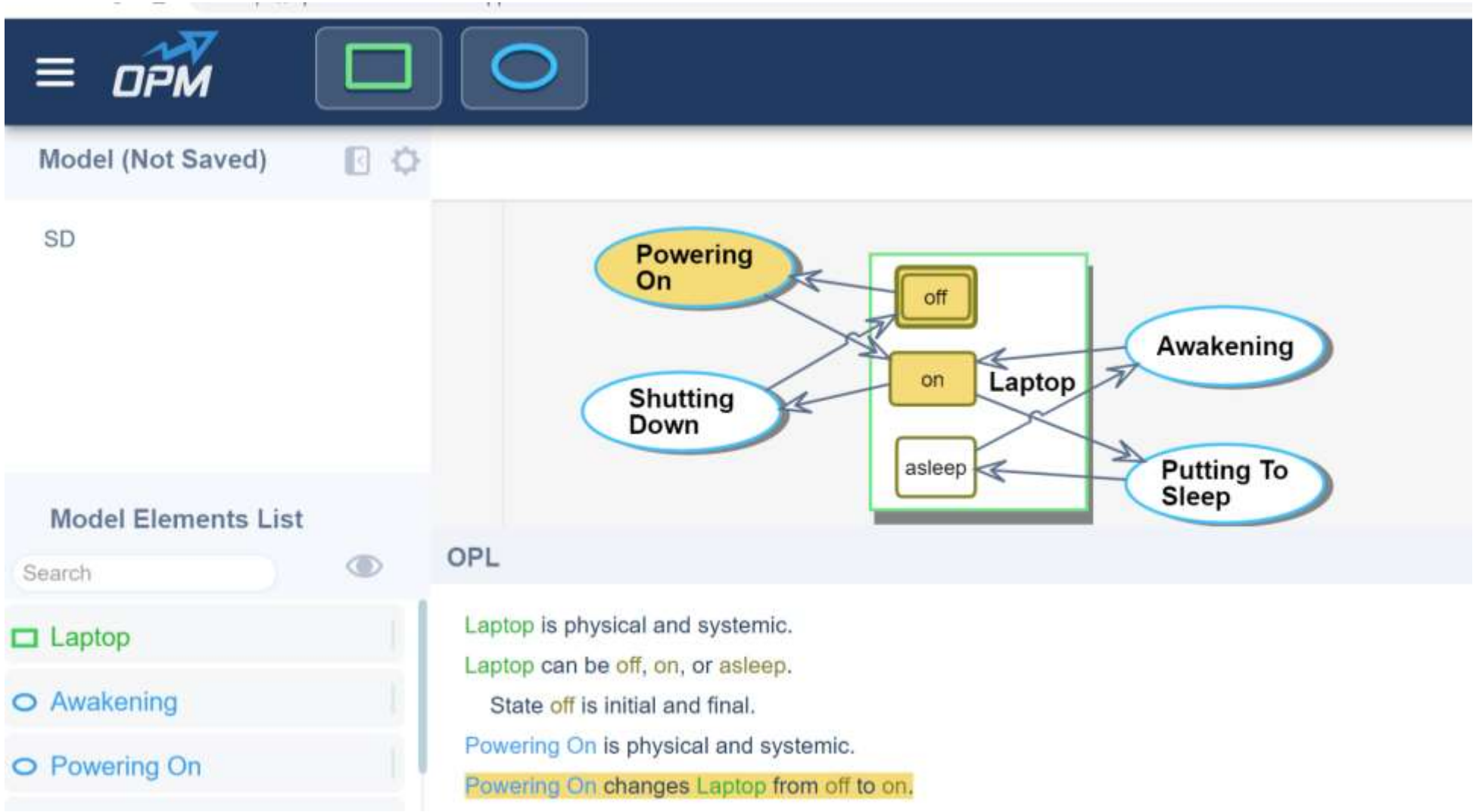
- *A* **process** *affects* an **object** *by changing its* **state.**
  - *Hence,* **objects** *must be* **stateful** *– they must have* **states.**

State: A situation an object can be at during its lifetime.

At each point in time, the object is

- at one of its states, or
- in transition from an ***input*** *state*
  - the input to the affecting process
- to its ***output*** *state*
  - the output of that process

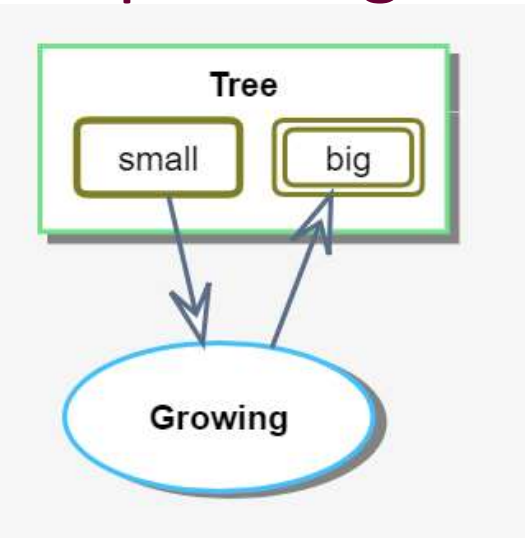
State transitions example



The graphics-text equivalence OPM principle

Any model fact expressed **graphically** in an OPD is also expressed **textually** in the corresponding OPL paragraph.

Each modality is reconstructible from the other.



Tree is physical and systemic.

Tree can be small or big.

State small is initial.

State big is final.

Growing is physical and systemic.

Growing changes Tree from small to big.

This bimodal representation caters to the dual channel cognitive assumption (Mayer, 2010)

Question 9: What are the two major aspects of any system?

- ***Structure*** – the static aspect:
  What the system is made of.
  - ***Objects*** *and relations among them*
  - ***Time-independent***
- ***Behavior*** – the dynamic aspect:
  How the system changes over time.
  - ***Processes*** *and how they transform objects*
  - ***Time-dependent***
- ***Time*** is the **discriminating** factor!

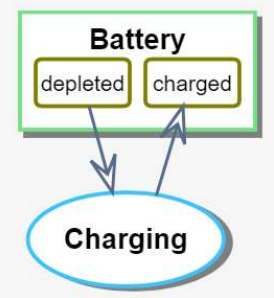
So far we discussed systems in general, without distinguishing
natural systems from
man-made ones

However, there is a fundamental difference between the two

Question 10: What third aspect is specific to human-made systems?

- ***Function*** – the utilitarian, subjective, benefit-providing aspect:
  - ***Why*** is the system built?
  - ***For whom*** is the system built?
  - ***Who*** benefits from operating the system?

- Function can be defined as any pair of **process** and its **operand**
  - e.g., **Battery Charging**; **Blood Pumping**
  - Natural systems may be beneficial or detrimental.

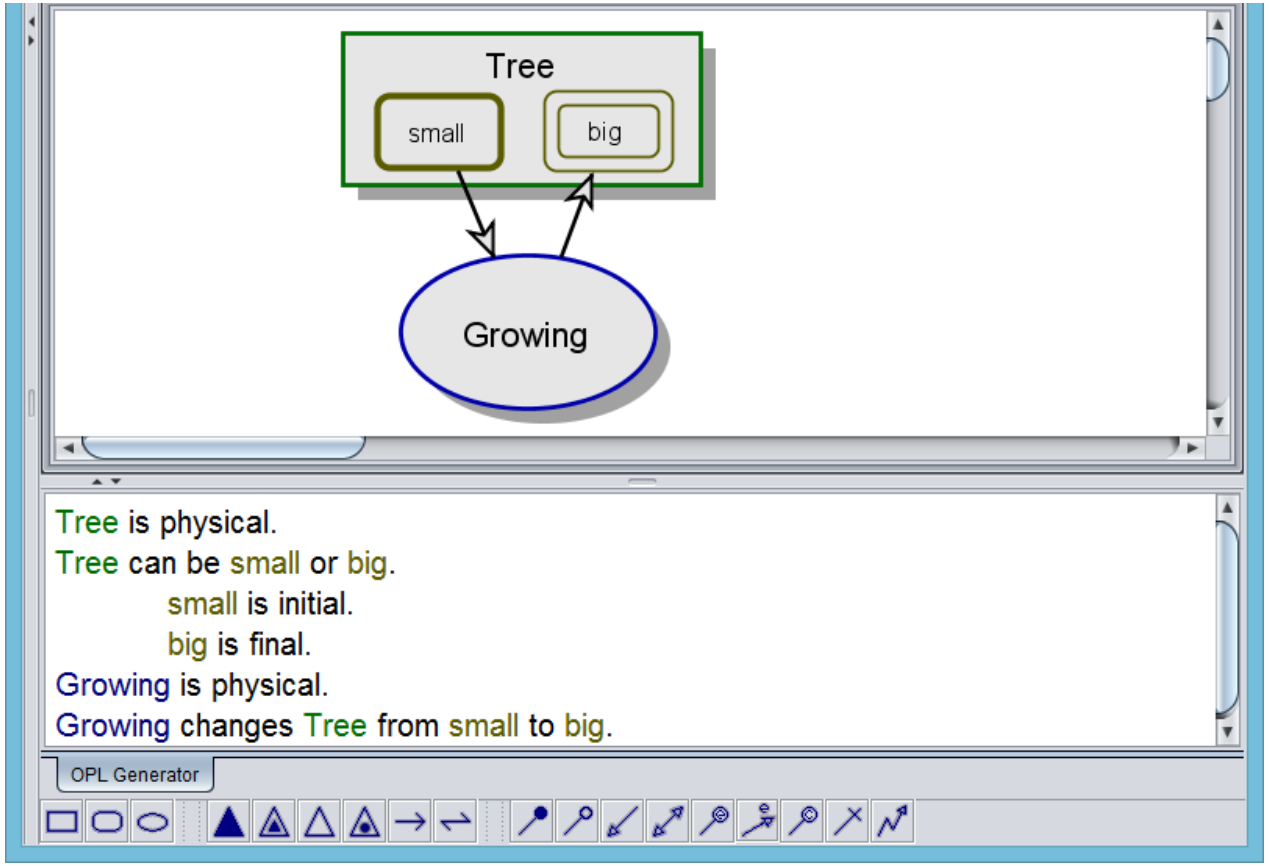


Charging changes Battery from depleted to charged.

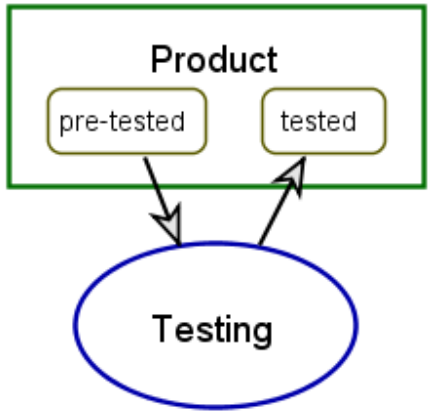
The graphics-text equivalence OPM principle

Any model fact expressed graphically in an OPD is also expressed textually in the corresponding OPL paragraph.

Caters to the dual channel cognitive assumption (Mayer, 2010)



- A process (even a physical one) is a cognitive pattern, in which we:
  - compare an object existence, or its state, in time points in the past vs. now, and
  - use this data to create a mental picture of the transformation the object undergoes.
- Only the objects involved in a physical process can be "touched"



The Object-Process Theorem

Stateful **objects**, **processes**,

and **relations** among them

constitute a **universal ontology.**

Caveat: May not apply to quantum-scale systems
(e.g., an electron may be both an object and a process)

Two Complementary Proofs:

1. Theoretical, based on logic and set theory
2. Empirical, based on many examples from many domains; no counter example found

The Things Axiom

The set $\mathbf{T}$ of all things (or entities, or concepts) in the universe is the union of two disjoint sets:

stateful objects Ω and processes Π:

$$\mathbf{T} = \Omega \cup \Pi; \Omega \cap \Pi = \phi$$

Every thing is either an object or a process

The Things-Relations Axiom

Representing any system Σ requires a set of things **T** and relations among them, **R**:

$$\mathbf{Rep}\ \Sigma \rightarrow \mathbf{T} \wedge \mathbf{R}$$

The Relations Lemma

- Let $\Re_s$ be the set of all the structural relations: relations
  between two objects, $\Re_{s(\Omega)}$, or
  between two processes $\Re_{s(\Pi)}$:

$$\Re_S = \Re_{S(\Omega)} \cup \Re_{S(\Pi)}$$

  Where

$$\Re_{S(\Omega)} = \{\rho \mid \omega_i \rho \omega_j;\ \omega_i, \omega_j \ni \Omega\}$$
$$\Re_{S(\Pi)} = \{\rho \mid \pi_i \rho \pi_j;\ \pi_i, \pi_j \ni \Pi\}$$

The Relations Lemma (cont.)

- Let $\Re_P$ be the set of all the procedural relations: relations between an object and a process:

$$\Re_P = \{\rho \mid \omega_i \rho \pi_j;\ \omega_i \ni \Omega,\ \pi_j \ni \Pi\}$$

Then the set of all possible relations $\Re$ between things in the universe is $\Re = \Re_S \cup \Re_P$

Or: **Any relation is either structural or procedural.**

Relations Lemma Proof

According to the Things Axiom, every thing is either an object or a process: $\mathrm{T} = \Omega \cup \Pi$

Therefore, relations can only exist between

(1) Two objects: These are in $\Re_{S(\Omega)}$,

(2) Two processes: These are in $\Re_{S(\Pi)}$,

(3) An object and a process: These are in $\Re_P$

Therefore, $\Re = \Re_{S(\Omega)} \cup \Re_{S(\Pi)} \cup \Re_P$

Substituting $\Re_S = \Re_{S(\Omega)} \cup \Re_{S(\Pi)}$

we get: $\Re = \Re_S \cup \Re_P$ QED

The Object-Process Theorem

Stateful objects, **processes**, and **relations** among them constitute a **necessary** and **sufficient universal ontology.**

Or: It is possible to **specify** any **system** in the universe using **stateful objects**, **processes**, and **relations** among them.

Caveat: May not apply to quantum-scale systems

Theoretical Proof Part 1 – Necessity

We need to show:

Stateful objects, **processes**, and **relations** among them are a necessary universal ontology.

In other words:

A set of **stateful objects** Ω,
a set of **processes** Π, and
a set of **relations** $\Re$ among them
are required to specify any system Σ in any domain in the universe.

Symbolically: $\text{Rep } \Sigma \rightarrow \Omega \wedge \Pi \wedge \Re$

$\text{Rep } (\Sigma) \rightarrow \mathsf{T} \wedge \Re$

Theoretical Proof Part 1 – Necessity

System aspect representation

Representing any system Σ requires representing its two aspects:

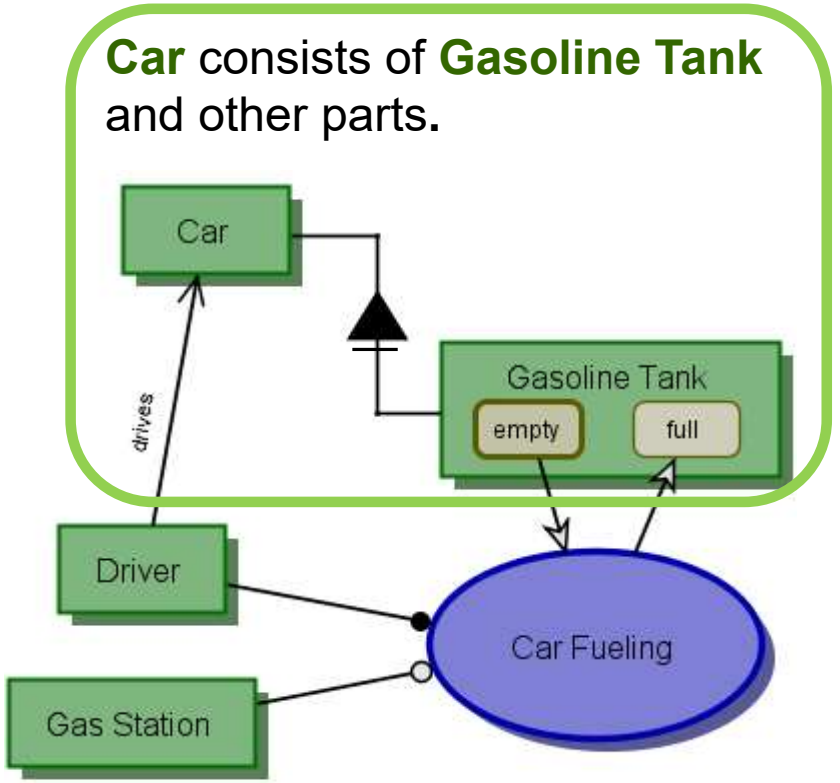
Structure, $S(\Sigma)$ and

Behavior, $B(\Sigma)$

Rep $\Sigma \rightarrow S(\Sigma) \wedge B(\Sigma)$

System structure representation

Representing S(Σ) (the system's structure) requires stateful objects Ω and structural relations R_s to link them:

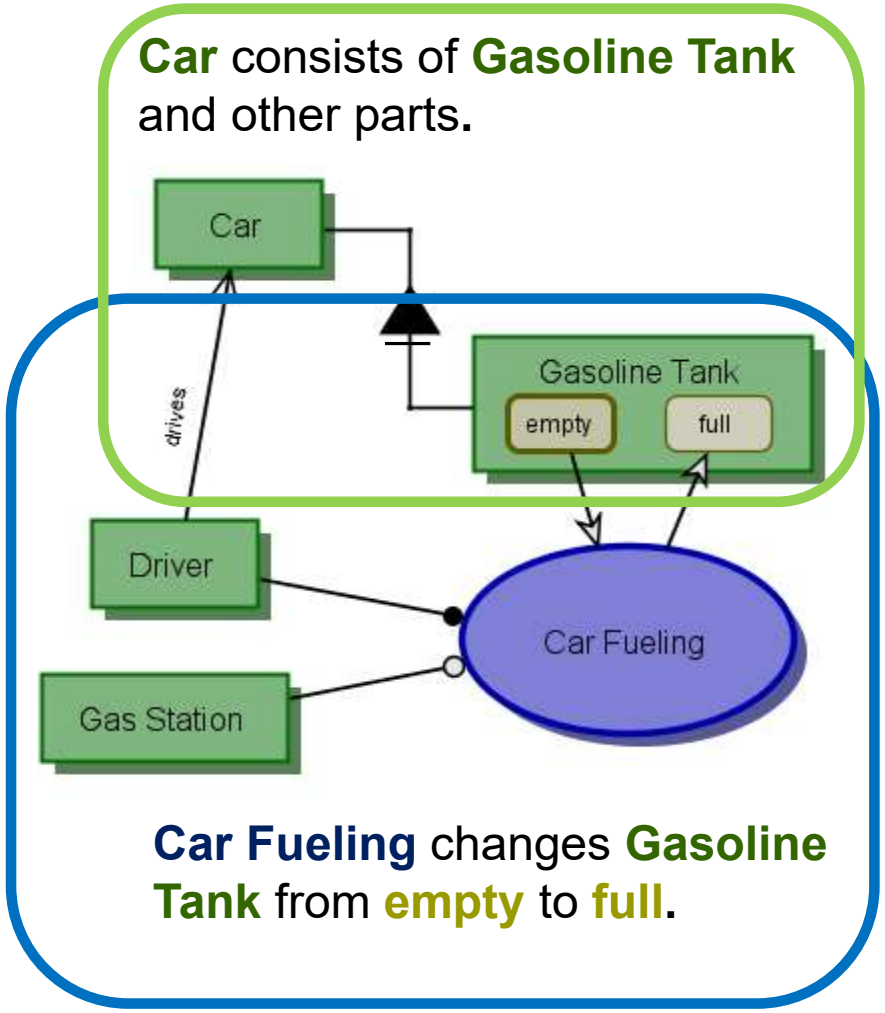


$$S(\Sigma) = \Omega \cup \Re_S$$

System behavior representation

Representing B(Σ) (the system's behavior) requires processes Π and procedural relations $\Re_P$ to link them to the objects they transform:

$$B(\Sigma) = \Pi \cup \Re_P$$



Substitution

We had:

$$\text{Rep } \Sigma \rightarrow S(\Sigma) \wedge B(\Sigma)$$

Substituting

$$S(\Sigma) = \Omega \cup \mathfrak{R}_S$$
$$B(\Sigma) = \Pi \cup \mathfrak{R}_P$$

We get:

$$\text{Rep } \Sigma \rightarrow (\Omega \cup \mathfrak{R}_S) \wedge (\Pi \cup \mathfrak{R}_P)$$

Rearranging terms (thanks to commutativity of union and conjunction) we get:

$$\text{Rep } \Sigma \rightarrow (\Omega \cup \Pi) \wedge (\Re_S \cup \Re_P)$$

The Things Axiom states that:

$$T = \Omega \cup \Pi$$

So the first term is the set of Things T and the second – the Relations R:

$$\text{Rep } \Sigma \rightarrow T \wedge \Re$$

Theoretical Proof Part 2 - Sufficiency

Stateful objects and **processes** are sufficient to specify any thing in any system, because:

- Anything that ***exists*** or ***might exist*** can be specified in terms of **stateful objects** and **relations** among them.
- Anything that ***happens*** or ***might happen*** to any stateful object can be specified in terms of **processes** and **relations** between the processes and the stateful objects they transform. **Q.E.D.**

Why does this make sense?

The universe is four-dimensional:

- Three spatial dimensions: x, y, z, for the static aspect
- One temporal dimension: t for the dynamic aspect

- Objects are **static** only in a snapshot (slice) of time t.
- To describe the **dynamic** aspect we must use t and show how objects change over time – these are the processes.

Empirical Proof of the Object-Process Theorem

Stateful objects, processes, and relations among them constitute a necessary and sufficient universal ontology.

If the ontology is universal, it must be able to model systems in any domain.

The empirical proof: Providing evidence of successful models from various, unrelated domains.

Molecular systems biology

Conceptual Modeling in Systems Biology Fosters Empirical Findings: The mRNA Lifecycle

Dov Dori, Mordechai Choder

Published: September 12, 2007 • DOI: 10.1371/journal.pone.0000872

Figures

Conceptual Model-Based Systems Biology: Mapping Knowledge and Discovering Gaps in the mRNA Transcription Cycle

Judith Somekh, Mordechai Choder, Dov Dori

Published: December 20, 2012 • DOI: 10.1371/journal.pone.0051430

Figures

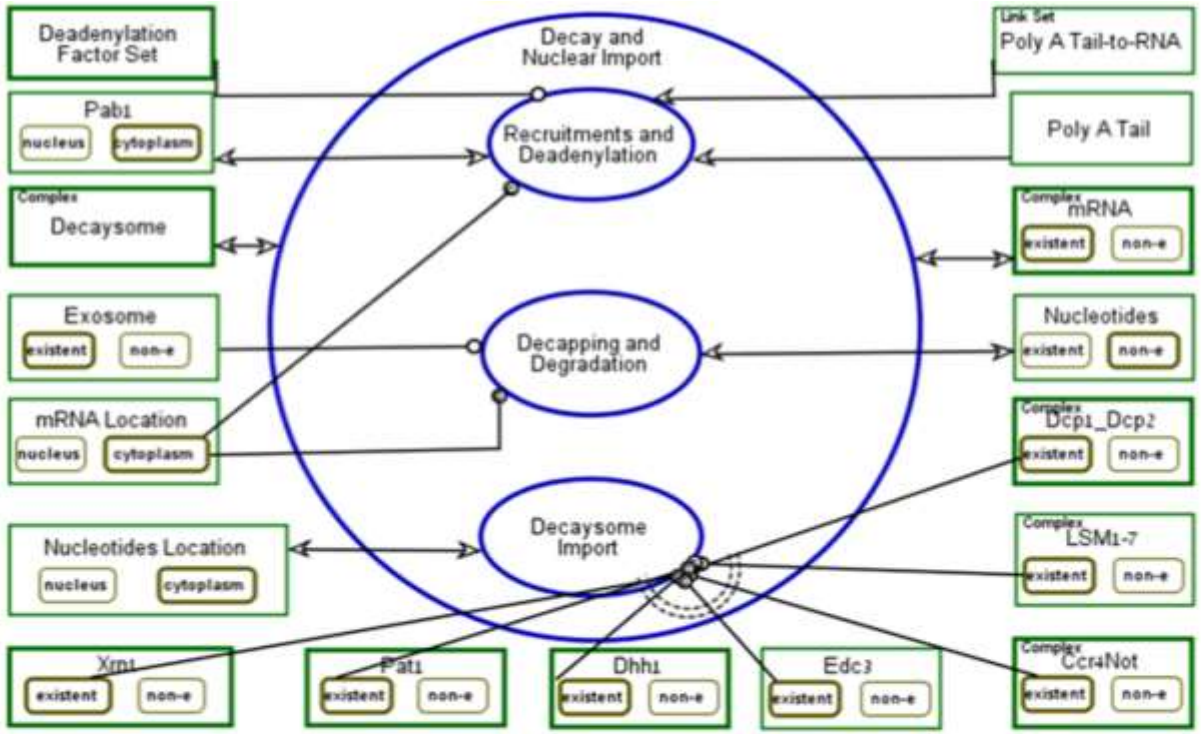
Conceptual Modeling of mRNA Decay Provokes New Hypotheses

Judith Somekh, Gal Haimovich, Adi Guterman, Dov Dori, Mordechai Choder

Published: September 25, 2014 • DOI: 10.1371/journal.pone.0107085

Figures

A



B

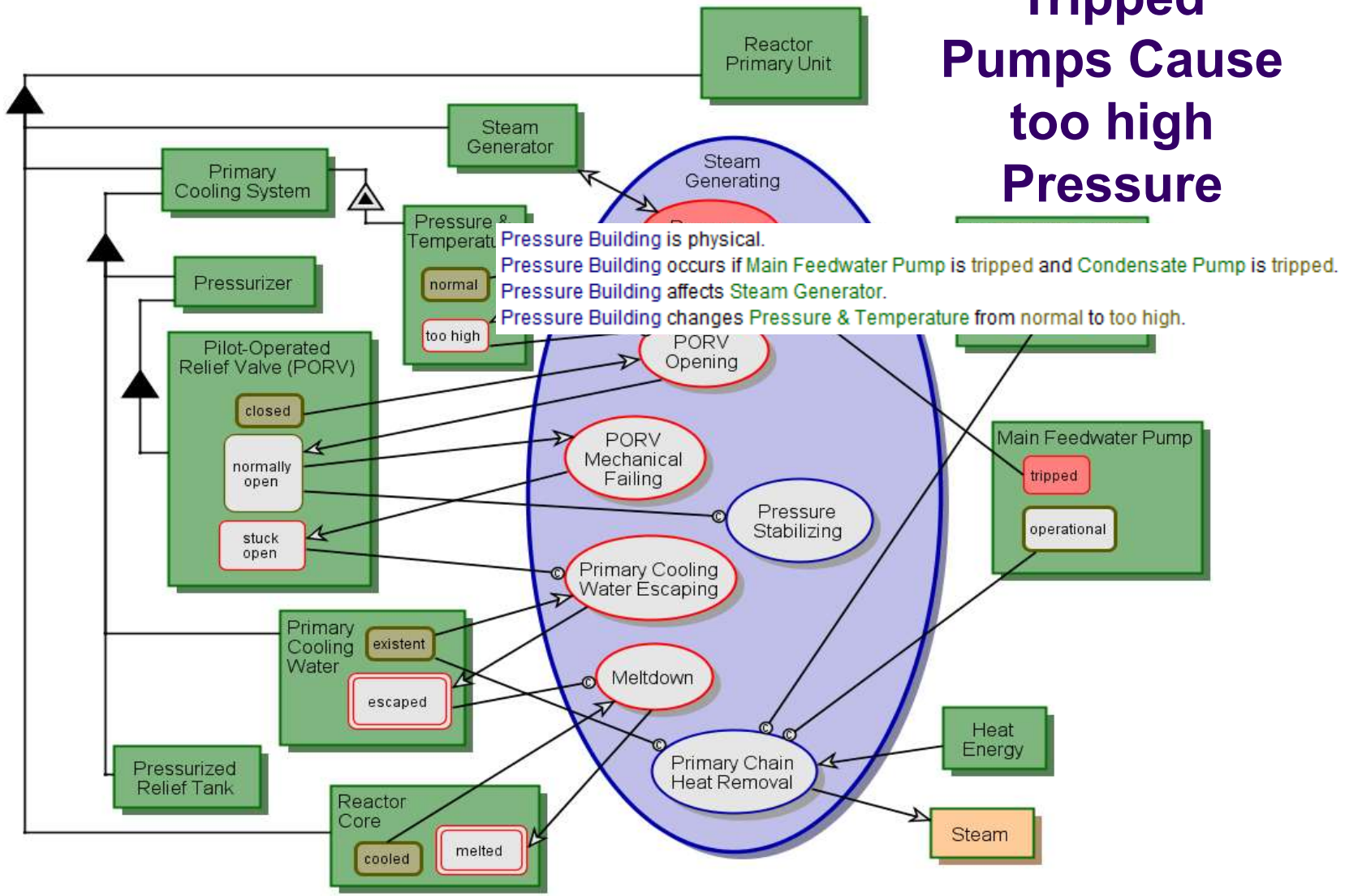
Ccr4Not can be existent or non-e. existent is initial. Ccr4Not plays the role of Complex. Dcp1_Dcp2 can be existent or non-e. existent is initial. Dcp1_Dcp2 plays the role of Complex. Decaysome plays the role of Complex. Dhh1 can be existent or non-e. existent is initial. Edc3 can be existent or non-e. existent is initial. Xrn1 ca

Decay and Nuclear Import affects Decaysome and mRNA.
Decay and Nuclear Import zooms into Recruitments and Deadenylation, Decapping and Degradation, and Decaysome Import.
Recruitments and Deadenylation occurs if mRNA Location is cytoplasm.

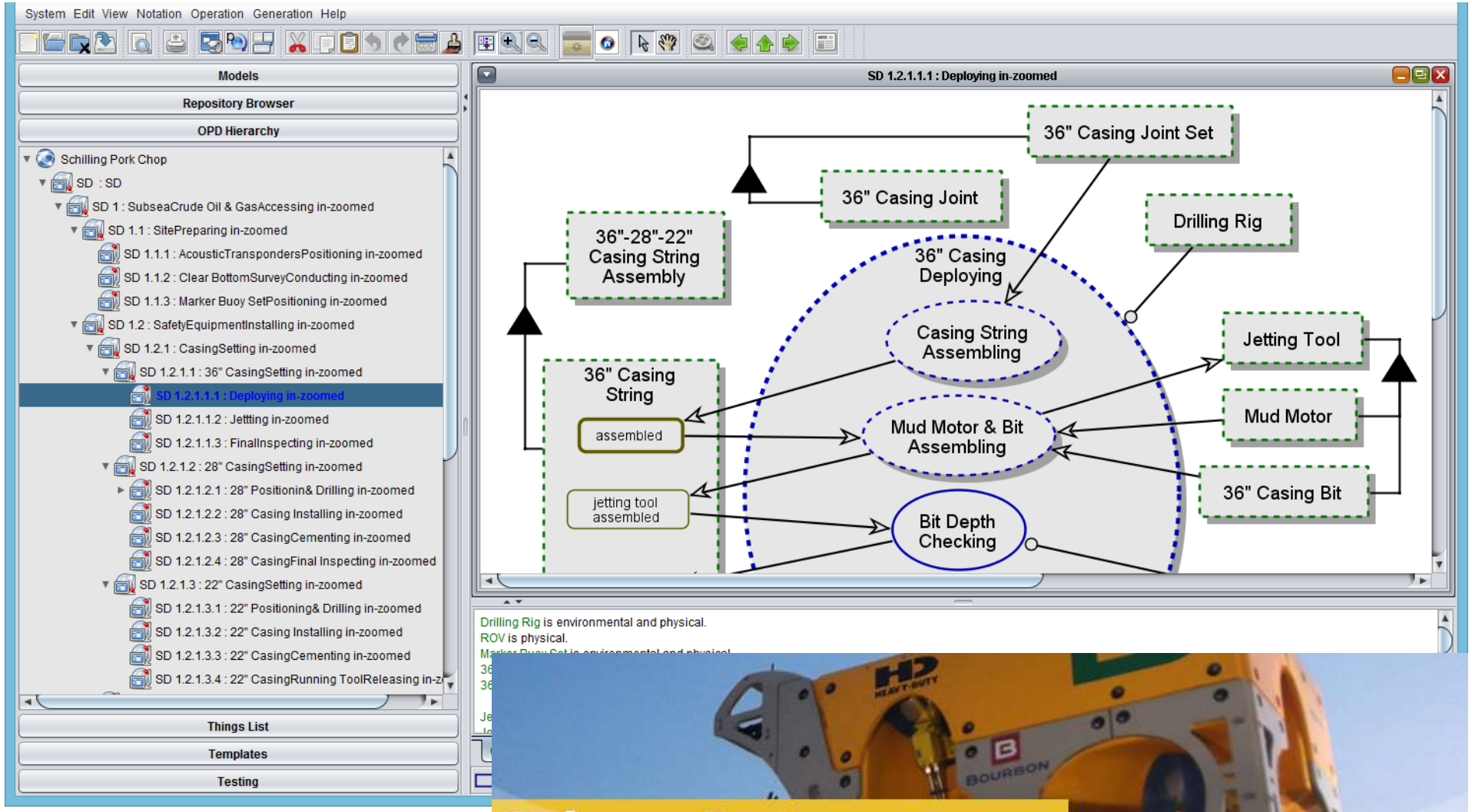
"Beyond the scientific value of these specific findings, this work demonstrates the value of the conceptual model as an in silico vehicle for hypotheses generation and testing, which can reinforce, and often even replace, risky, costlier wet lab experiments."

Nuclear reactor failure: The Three Mile Island Accident

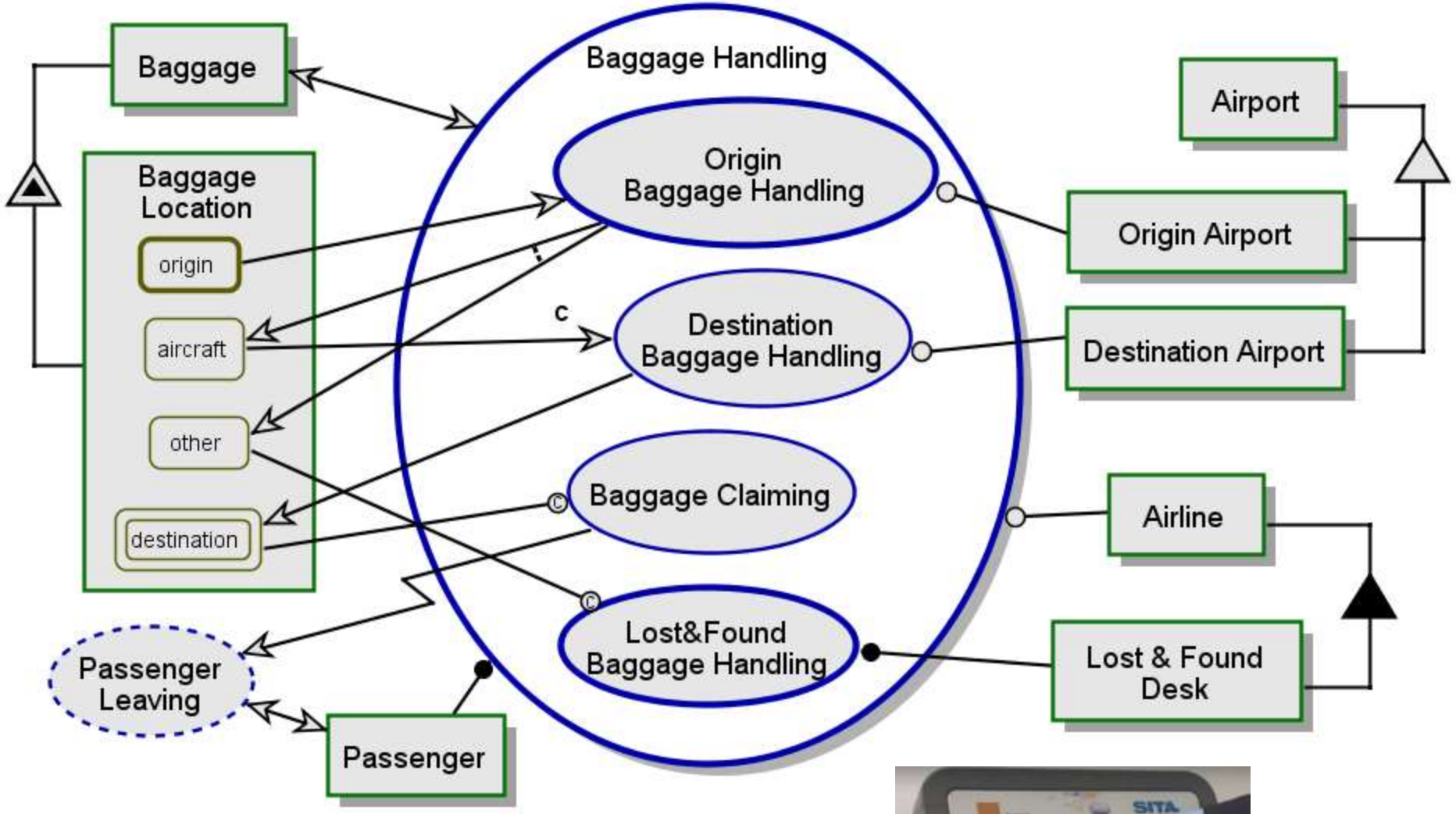
Tripped Pumps Cause too high Pressure



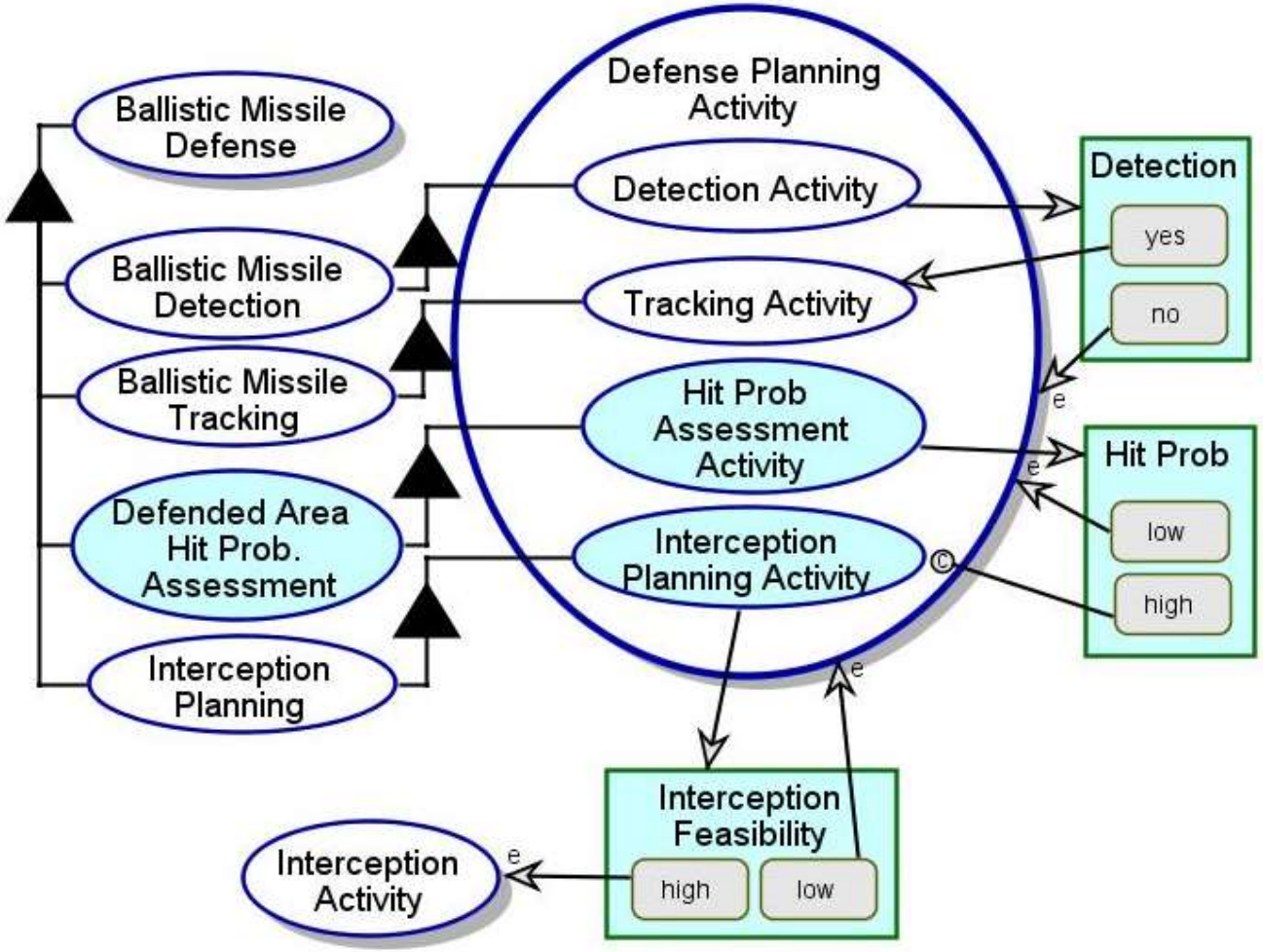
Offshore Oil Well Drilling



Airport Operations: Outgoing Passenger



Iron Dome – an Israeli ballistic missile defense system: 90% hit rate



Yaniv Mordecai and Dov Dori, Evolving System Modeling: Facilitating Agile System Development with Object-Process Methodology. *SysCon 2015 ,9th Annual IEEE International Systems Conference*, Vancouver, Canada, April 13-16 2015.

Sample of engineering domains in which OPM has been used

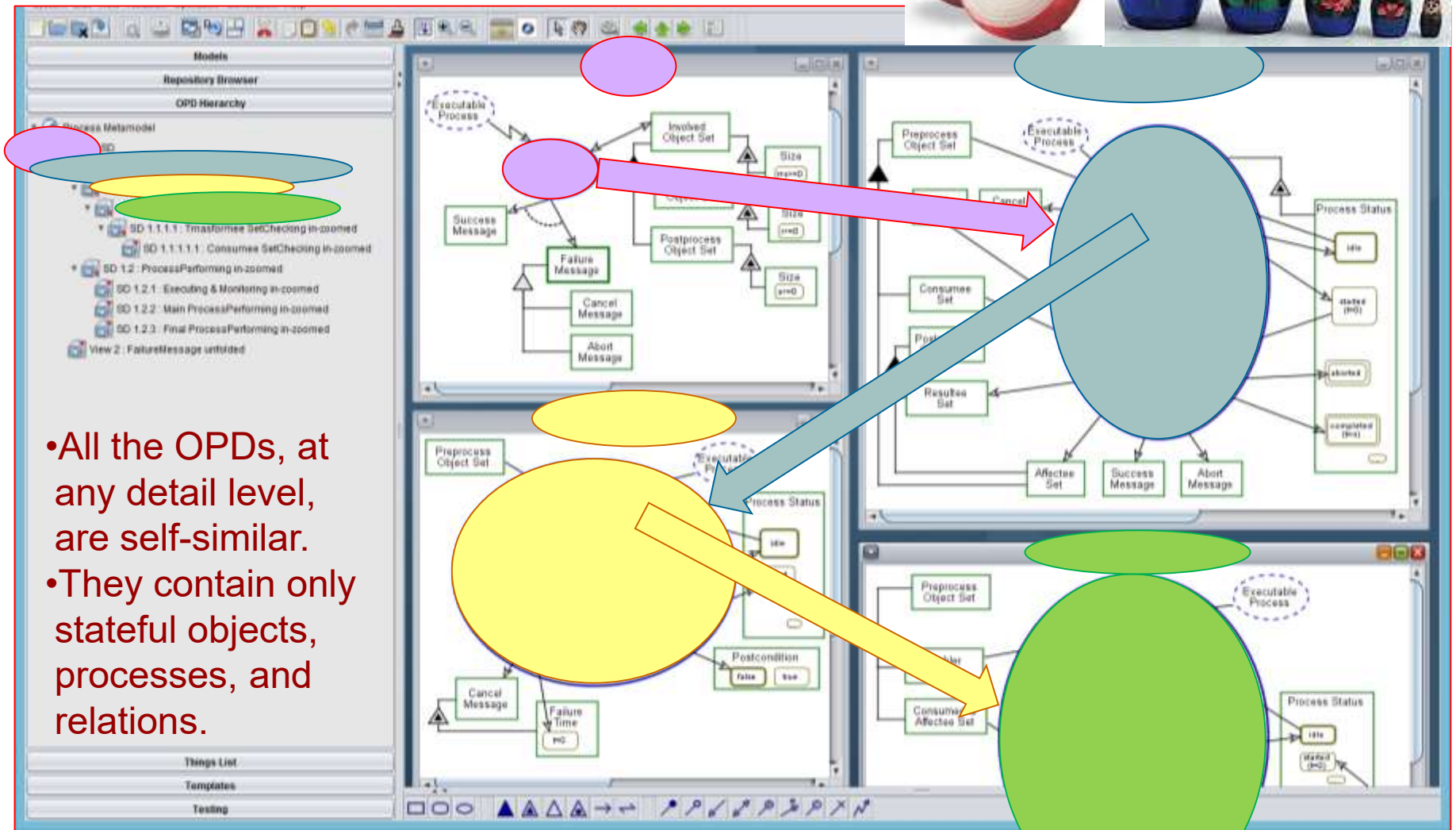
- **Complex, Interconnected, Large-Scale Socio-Technical Systems.** *Systems Engineering* 14(3), 2011.
- **Networking Mobile Devices and Computers in an Intelligent Home.** *International Journal of Smart Home* 3(4), pp. 15-22, October, 2009.
- **Multi-Agent Systems.** *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 40 (2) pp. 227-241, 2010.
- **Semantic Web Services Matching and Composition.** *Web Semantics: Science, Services and Agents on the World Wide Web*. 9, pp. 16-28, 2011.
- **Project-Product Lifecycle Management.** *Systems Engineering*, 16 (4), pp. 413-426, 2013.
- **Model-Based Risk-Oriented Robust Systems Design.** *International Journal of Strategic Engineering Asset Management*, 1(4), pp. 331-354, 2013.
- **Medical Robotics and Miscommunication Scenarios.** An Object-Process Methodology Conceptual Model. *Artificial Intelligence in Medicine*, 62(3) pp. 153-163, 2014.
- **Modeling Exceptions in Biomedical Informatics**. *Journal of Biomedical Informatics 42*(4), pp. 736-747, 2009.

Complexity Management with OPM

- Systems are inherently complex.
- To alleviate this complexity, in OPM, it is managed by detail decomposition through three refinement-abstraction:
  - In-zooming – Out-zooming
  - Unfolding – Folding
  - State expression – suppression.

In-zooming – Out-zooming Example

Process Performance Controlling
- a metamodel from ISO 19450



OPM Complexity Management Benefits

- There is no limit on the level of complexity of the system being modeled:
  - One can specify system structure and behavior at any level of detail by recursively in-zooming.
- Catering to the cognitive limited capacity:
  - Each diagram is not overly complicated.
- All the diagrams are “aware” of each other:
  - All OPDs are partial views of the same system.
  - Any change in one diagram is propagated to all the other relevant ones.
  - All OPDs, regardless of the level of detail, are self-similar: They all use the same compact universal ontology

OPM – Object-Process Methodology

- A language and methodology for modeling complex systems of any kind
- Recognized as ISO 19450
- Based on the minimal universal ontology of
  - stateful objects
  - processes that transform objects by:
    - Creating new objects
    - Consuming existing ones, or
    - Changing their states

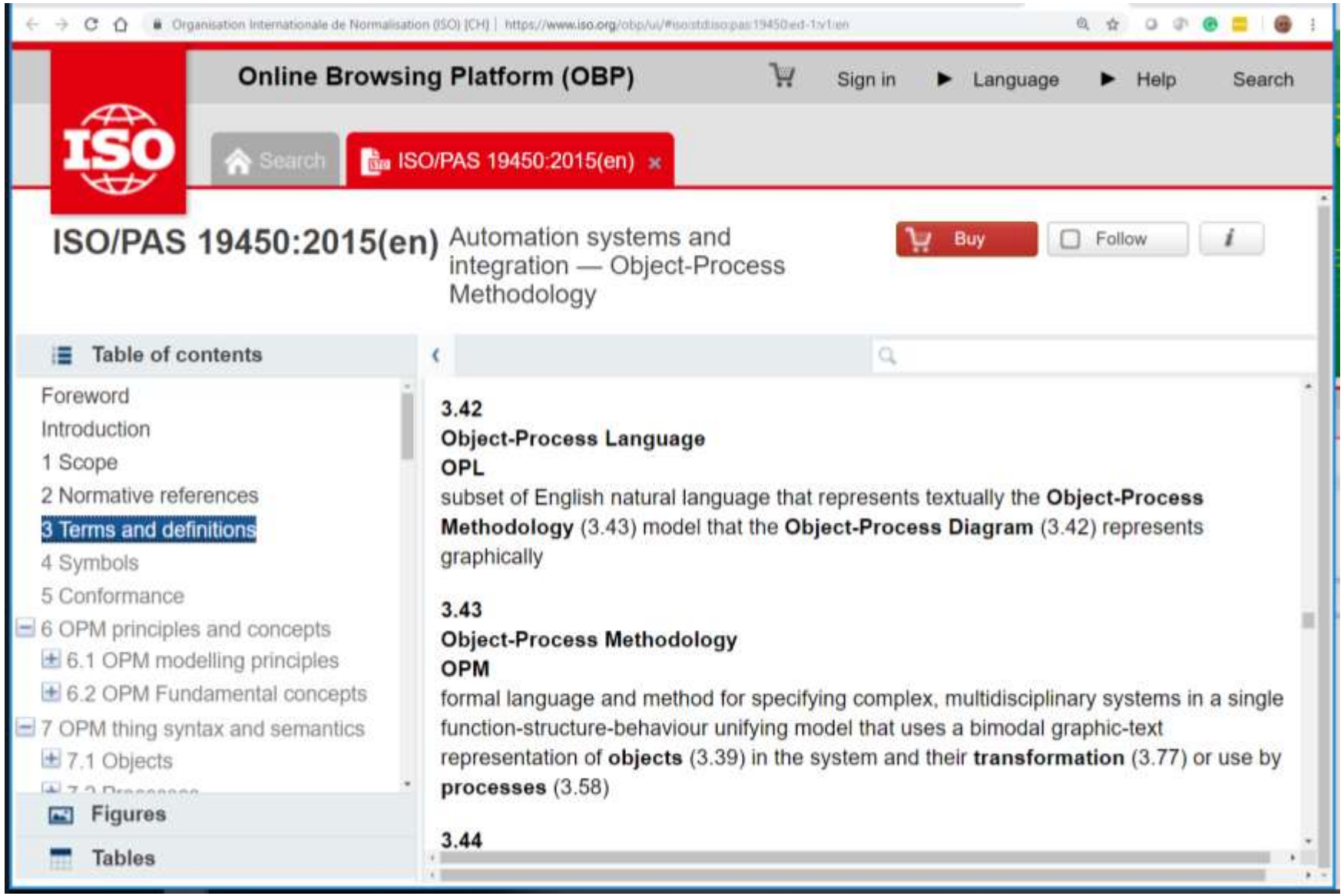
OPM – Highlights

- Bi-modal visual-textual presentation:
  - OPD – Object-Process Diagram
  - OPL – Object-Process Language
- Complexity management:
  - Things – objects and processes – can be refined to any desired level of detail
  - Via refinement-abstraction mechanisms:
    - In-zooming – Out-zooming (primarily for processes)
    - Unfolding – Folding (primarily for objects)
    - Expressing – Suppressing (for states)

OPM – More Highlights

- Executable:
  - Objects can be computational (numbers, parameters)
  - Processes can perform math and embed code
  - The model can be seamlessly and concurrently executed both
    - conceptually-qualitatively and
    - concretely-qualitatively
  - In two modes:
    - Online live graphic mode: animation for visualization & debugging
    - Offline batch mode: numeric simulation for big data, statistical analysis

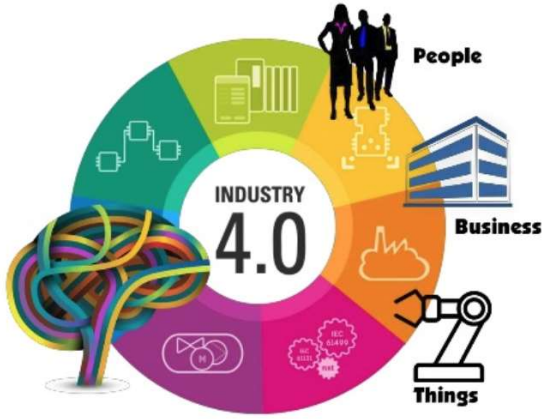
2015: OPM becomes ISO 19450



Industry 4.0

Industry 4.0 is the ultimate blend of hardware and software

Industry 4.0 mixes hardware and software to the extent that they are inseparable!

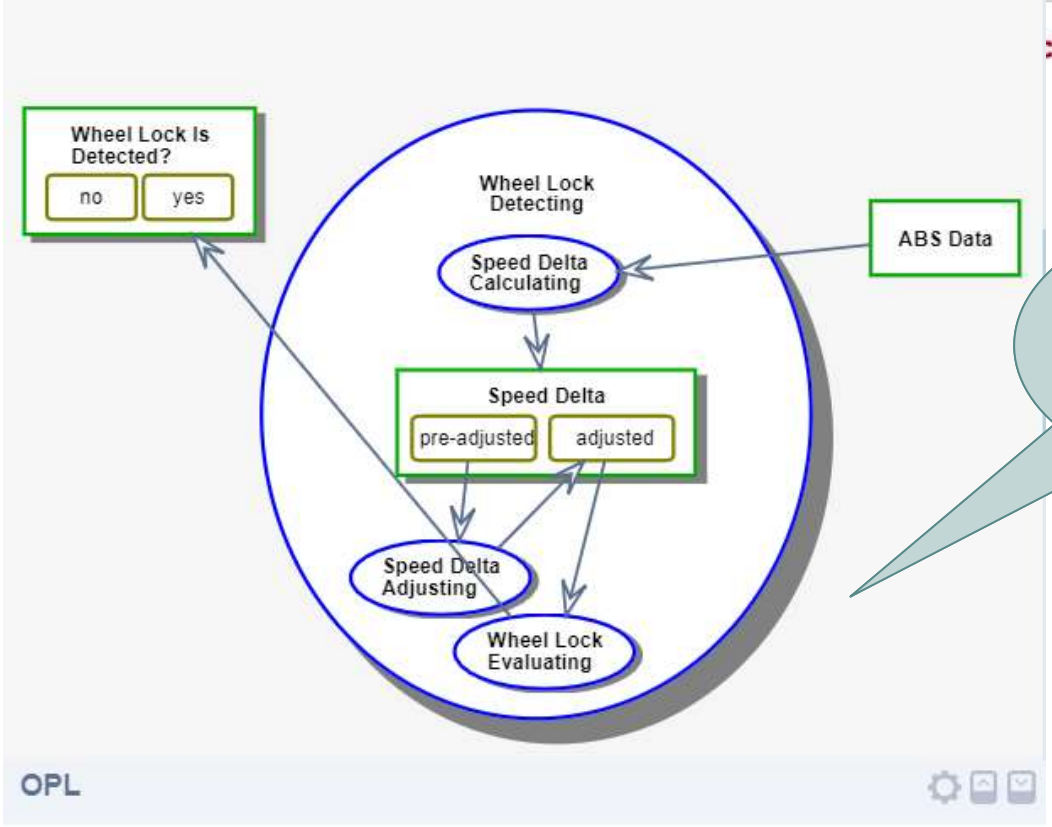


OPM is perfect for Industry 4.0

SOFTWARE **HARDWARE**

Smart **Robotics**
Smart **Manufacturing**
Data **Capture** (**sensors**) and analytics
Digital **Fabrication** (**3D Printing**…)
Cloud Computing in **Computer Farms**
Location & navigation services
Smart **Phones**
Internet of **Things**
Physical and Digital **Twins**
Autonomous **Transportation**

OPM = OPD + OPL



OPD - Object Process Diagram

OPL

Wheel Lock Detecting from SD1.1.1.1 zooms in SD1.1.1.1.3 into Speed Delta Calculating, Speed Delta Adjusting, and Wheel Lock Evaluating, as well as Speed Delta.

ABS Data is informatical.

Wheel Lock Is Detected? can be no or yes.

Speed Delta can be pre-adjusted or adjusted.

Speed Delta Calculating consumes ABS Data.

Speed Delta Calculating yields Speed Delta.

Speed Delta Adjusting changes Speed Delta from pre-adjusted to adjusted.

Wheel Lock Evaluating consumes adjusted Speed Delta.

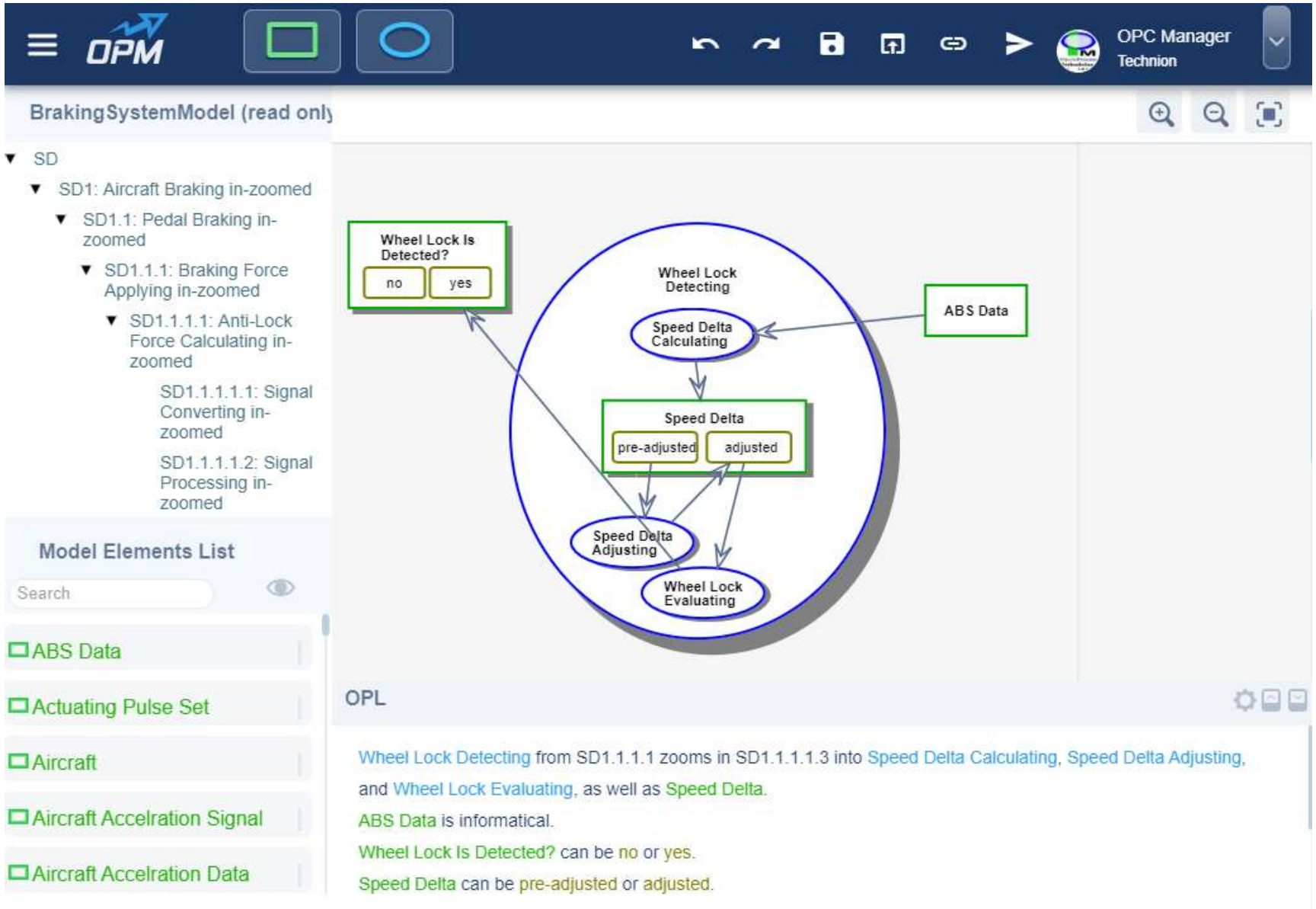
Wheel Lock Evaluating yields Wheel Lock Is Detected?.

OPL - Object Process Language

Bimodal graphics-text representation caters to dual channel processing

OPCloud

https://www.opcloud.tech/



OPCloud main features

Cloud-based application - ability to work from anywhere, any time

Collaborative and simultaneous work of multiple users, one editor at a time

Connectivity with to other systems (DOORS, ARAS, PTC...) using OSLC

Latest Web development technologies: Firebase, Angular, Rappid…

Correctness-by-construction via context sensitivity for choice of links and other features

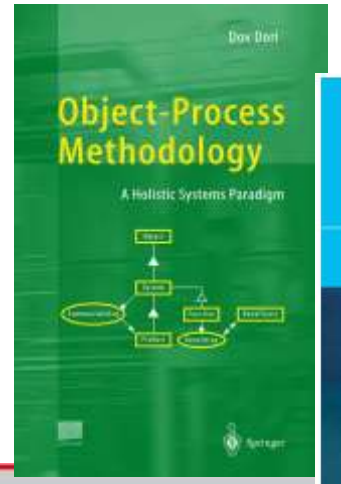
Backward compatibility for OPM models prepared with OPCAT

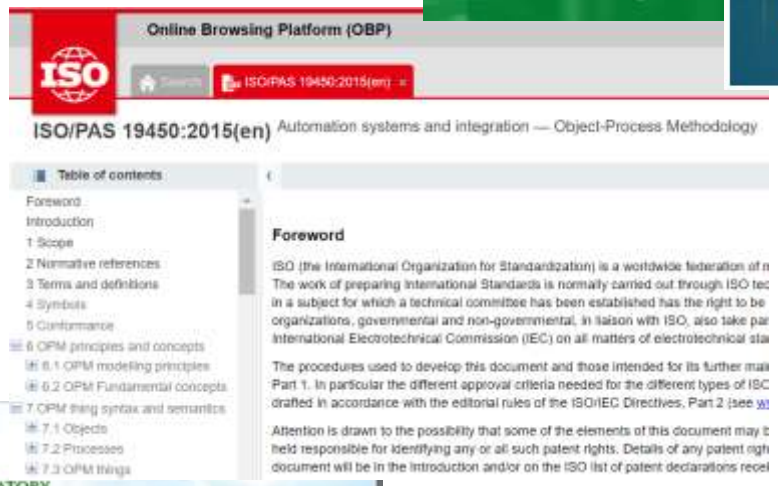
Where is OPM used?

- Automotive industry
- Aviation industry
- White appliances industry
- Energy sector
- Insurance sector
- Molecular biology
- Space agencies

OPM Resources

- Book (2002): Object-Process Methodology - A Holistic Systems Paradigm, Springer Verlag, Berlin, Heidelberg, New York, 2002.
- Book (2016): Model-Based Systems Engineering with OPM and SysML, Springer, New York.
- ISO Standard: ISO 19450 OPM
- edX Course: **https://www.youtube.com/watch?v=5ZhEreXtu0k**
- Website: Enterprise Systems Modeling Laboratory contains
  - journal & conference papers,
  - free **OPCAT** software, upcoming **OPCloud**
  - presentations
  - projects
  - more...
- OPCloud: https://www.opcloud.tech/





Model-Based Systems Engineering with OPM:

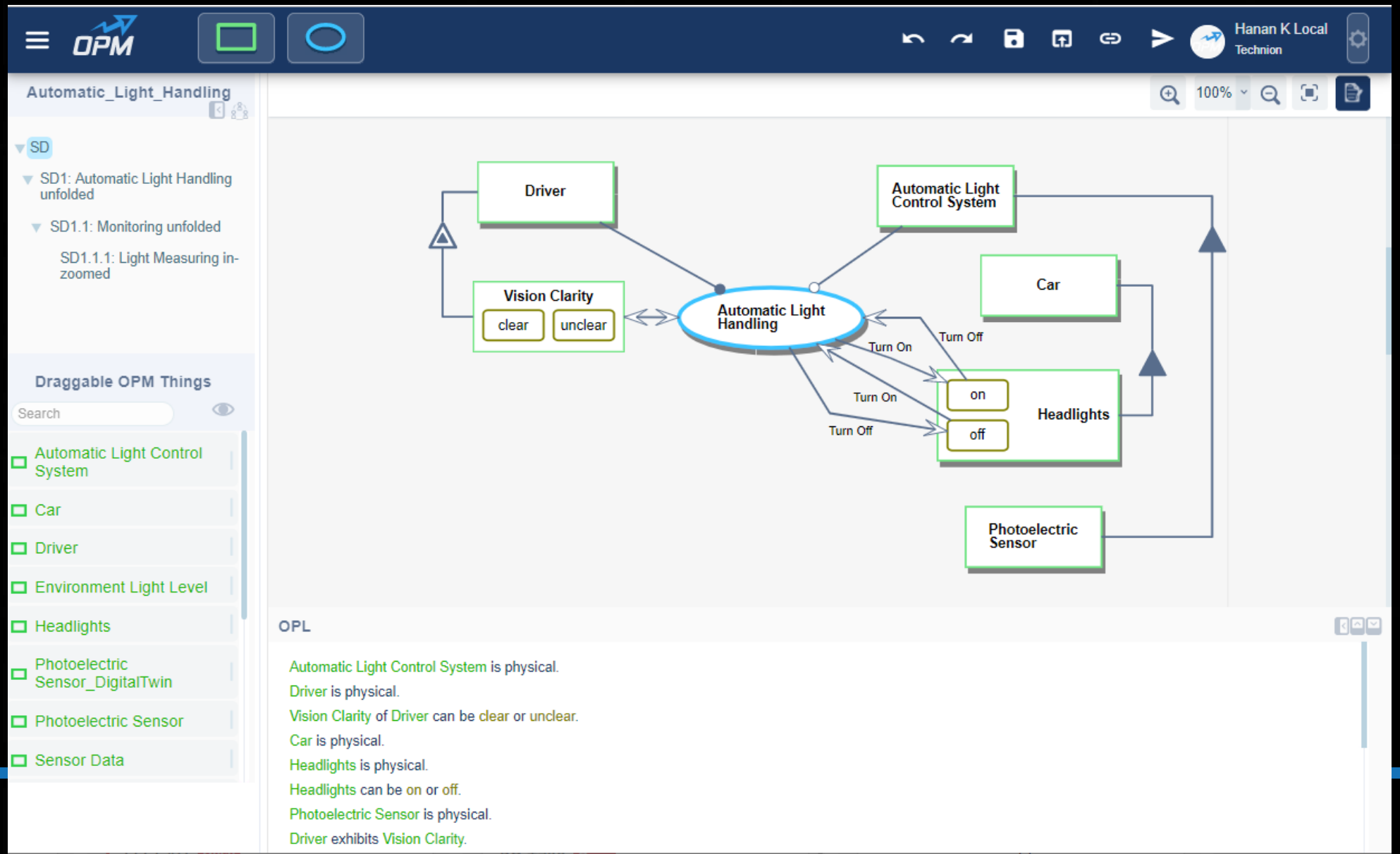
edX Course Series and Professional Certificate Program

Sample of engineering domains in which OPM has been used

- **Complex, Interconnected, Large-Scale Socio-Technical Systems.** *Systems Engineering* 14(3), 2011.
- **Networking Mobile Devices and Computers in an Intelligent Home.** *International Journal of Smart Home* 3(4), pp. 15-22, October, 2009.
- **Multi-Agent Systems.** *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 40 (2) pp. 227-241, 2010.
- **Semantic Web Services Matching and Composition.** *Web Semantics: Science, Services and Agents on the World Wide Web*. 9, pp. 16-28, 2011.
- **Project-Product Lifecycle Management.** *Systems Engineering*, 16 (4), pp. 413-426, 2013.
- **Model-Based Risk-Oriented Robust Systems Design.** *International Journal of Strategic Engineering Asset Management*, 1(4), pp. 331-354, 2013.
- **Medical Robotics and Miscommunication Scenarios.** An Object-Process Methodology Conceptual Model. *Artificial Intelligence in Medicine*, 62(3) pp. 153-163, 2014.
- **Modeling Exceptions in Biomedical Informatics**. *Journal of Biomedical Informatics 42*(4), pp. 736-747, 2009.

OPCloud – Implements OPM ISO 19450

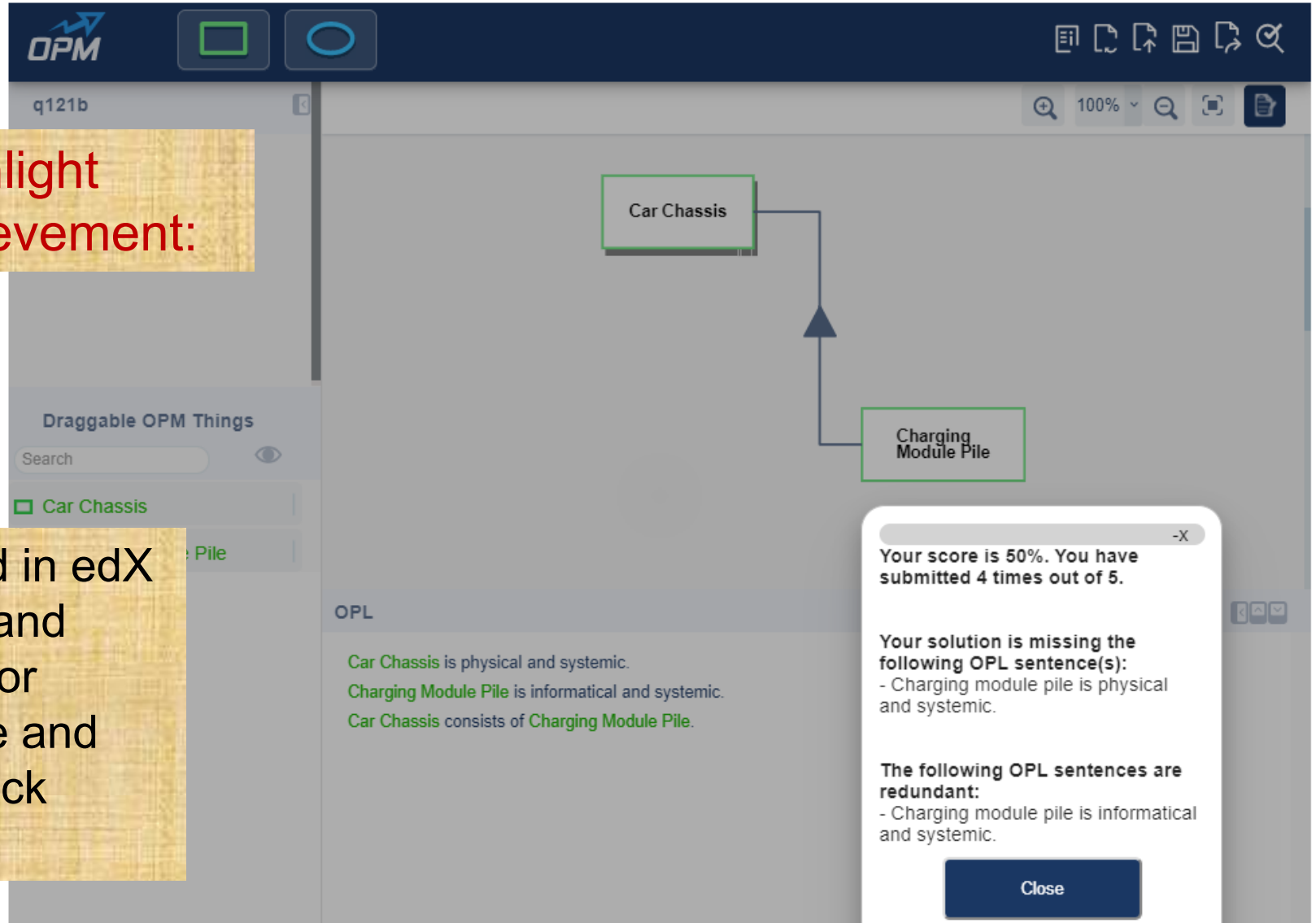
https://www.opcloud.tech/



MORTIF—Modeling with Real-Time Informative Feedback

Active Learning Highlight &Technological achievement:

- OPCloud embedded in edX
- Using LTI standard and
- a dedicated server for
- Immediate response and
- Constructive feedback



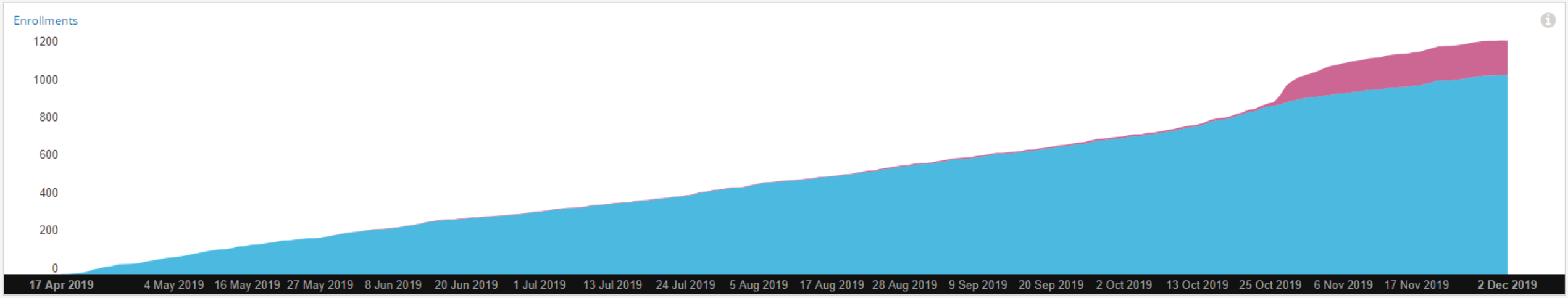
Student Enrolment – Basic Course

edX INSIGHTS Model-based Systems Engineering: Foundations Help DovDori

Course Home | ENROLLMENT | Activity | Demographics | Geography

Daily Learner Enrollment

How many learners are in my course?



Enrollment Metrics

1,329	1,238	19	183
Total Enrollment	Current Enrollment	Change in Last Week	Verified Enrollment

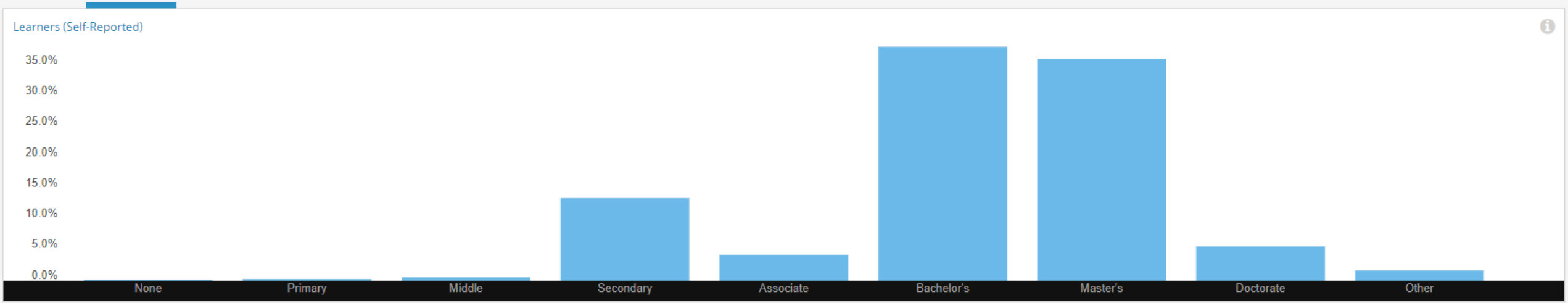
Students' Education – Basic Course (N=1329)

edX INSIGHTS Model-based Systems Engineering: Foundations Help DovDori

Course Home | ENROLLMENT | Activity | Demographics | Geography

Age | Education | Gender

What level of education do my learners have?



Education Metrics

14.3%	42.3%	41.7%
High School Diploma or Less	College Degree	Advanced Degree

MORTIF—Modeling with Real-Time Informative Feedback

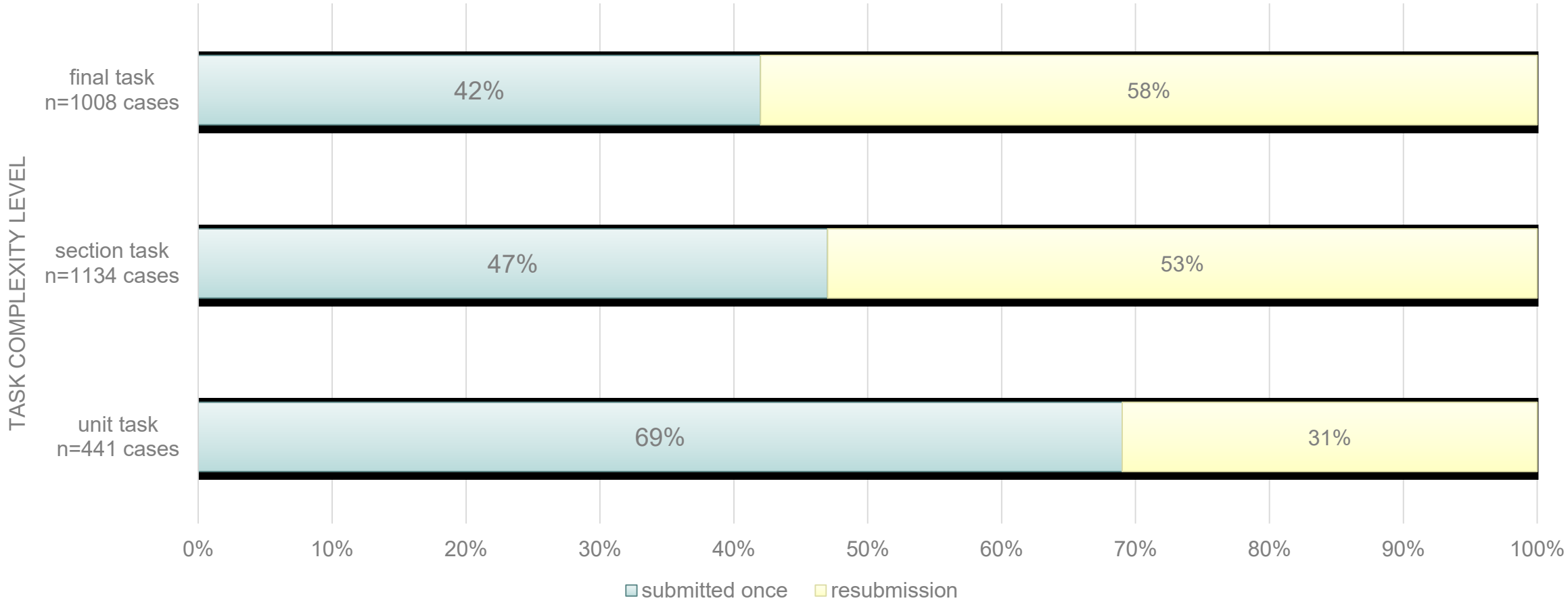
Preliminary Research Findings

Research Purpose

Examine:

- The usability of the MORTIF system
- Its contribution to the learning process

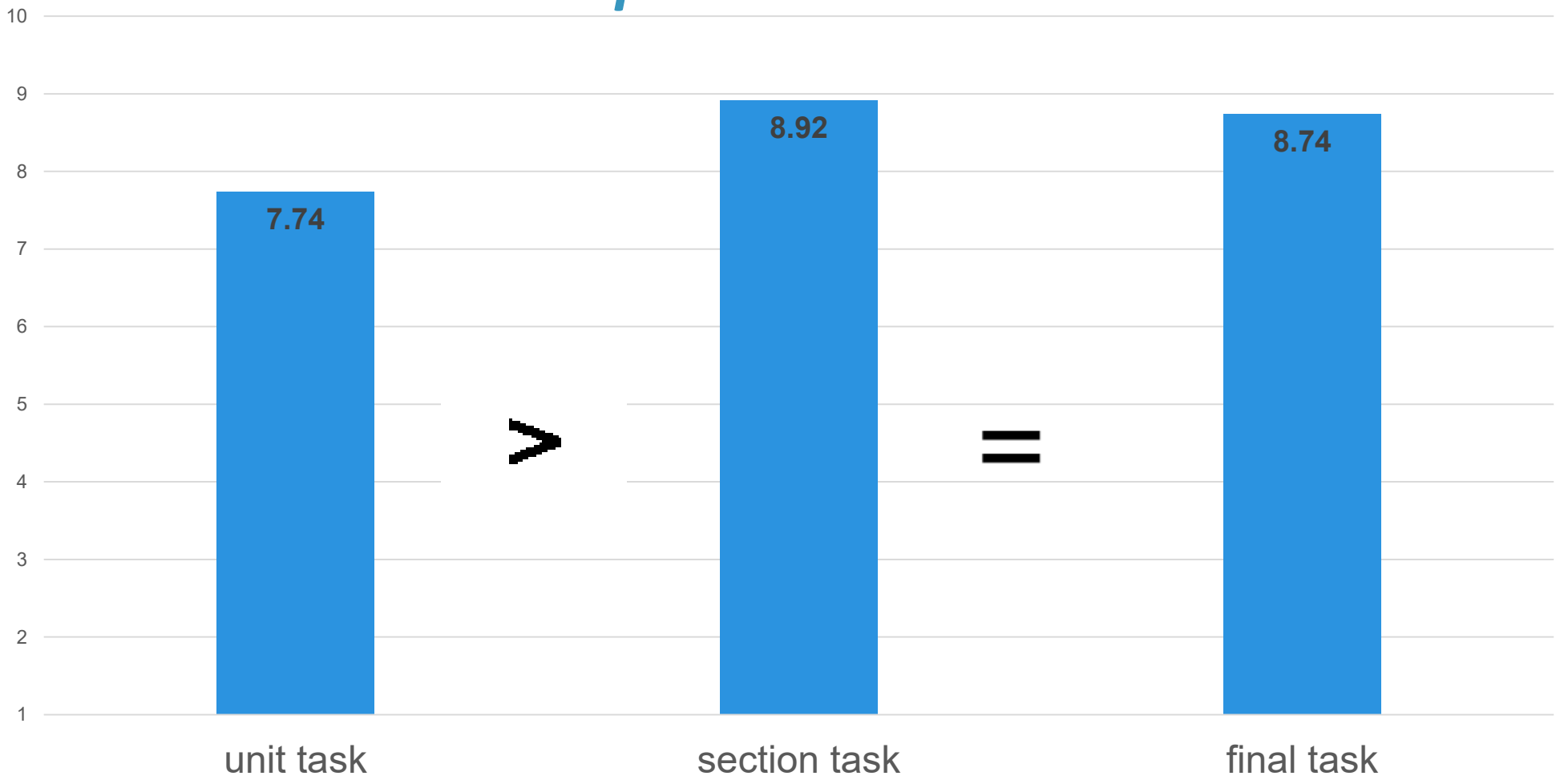
MORTIF-type assignments usability: Students' Preferences



$F(2, 110)=158.18, p<.001, \eta^2=.74$ ⇨ final > section > unit

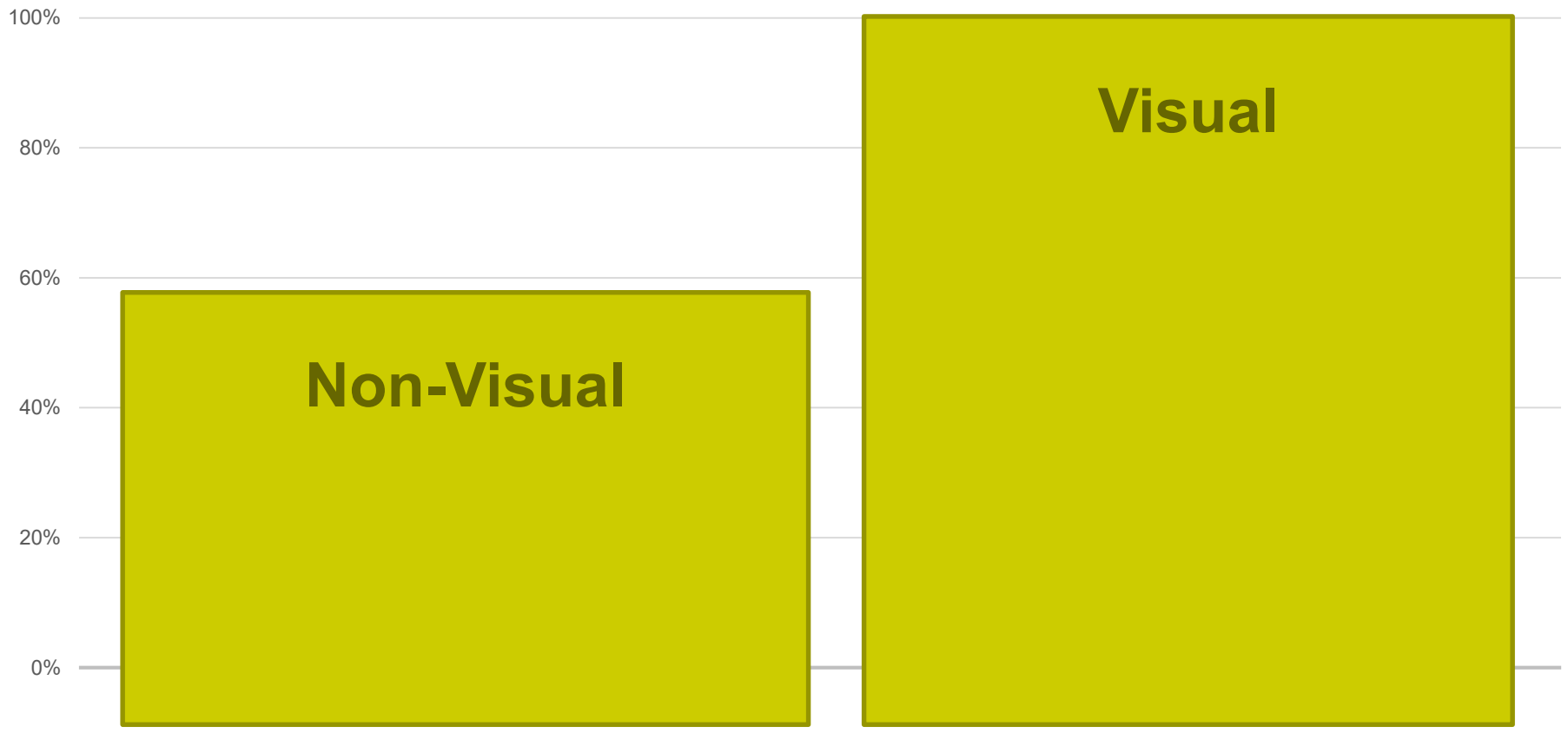
MORTIF-type assignments

Students' perceived contribution



$F(2, 74)=16.84, p<.001, \eta^2=.31$ ⇨ final & section > unit

MORTIF vs. other assignment types: Students' perceived contribution



MORTIF-type assignments

Learning style suitability

Learning style	Focus on	Student excerpt example	Frequency
Sequential vs. Global	information organizing preference	*Felt like a continuous problem that added more with each step.*	10
Sensing vs. Intuitive	information collecting preference	*The possibility to submit several times allowed a trial and error process.*	11
Active vs. Reflective	information processing	*Constructing the model really made the learning deeper than all the other problem types.*	59
Visual vs. Verbal	information presentation preference	*I prefer the graphics and visual problems.*	13

Thanks for listening!

Visit our Lab site;
http://esml.iem.technion.ac.il/

Experience OPCloud, Cloud-based OPM modeling:
https://www.opcloud.tech/

Contact us:

Tel: +972-77-8872441

OPCloud@technion.ac.il

Enterprise Systems Modeling Laboratory 121 Bloomfield, Technion - Israel Institute of Technology Haifa, 3200003 Israel

SysML and OPM – a brief comparison

Feature	SysML	OPM
Theoretical foundation	UML; Object-Oriented paradigm	Minimal universal ontology; Object-Process Theorem
Standard documentation number of pages	~1670=700 (UML Infrastructure) + 700 (UML Superstructure) + 270 (OMG SysML)	~180=100 (ISO 19450 main standard) + 80 (appendices)
Standardization body	OMG	ISO
Number of diagram kinds	9	1
Top-level concept	Block (UML object class)	Thing (object or process)
Complexity management guiding principle	Aspect-based decomposition	Detail-level-based decomposition
Hierarchical decomposition	In some diagram kinds	Yes
Number of symbols	~120	~20
Graphic modality	Yes	Yes
Textual modality	No	Yes
Built-in physical-informatical distinction	No	Yes
Systemic-environmental distinction	Partial (using boundaries)	Yes
Logical relations (OR, XOR, AND)	No	Yes
Probability modeling	No	Yes
Execution, animated simulation, validation and verification capability	Partial (in some tools for some diagram kinds)	Yes
Tool availability	Many, some free	Currently one free (OPCAT) from http://esml.iem.technion.ac.il/ Cloud-based tool under development