The Minimal Universal Ontology of Stateful Objects and Processes that Transform Them

OPM – ISO 19450

Dov Dori
Ontology Summit
Jan. 24 2018

Technion, Israel Institute of Technology
Massachusetts Institute of Technology
Ontology & Universal Ontology

Ontology: a set of concepts for describing a domain (industry, banking, military, botany, healthcare…) and systems within it.

Universal Ontology: a domain-independent set of concepts 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

Any thing either exists or happens – nothing else!
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)

Crashing  
Baking  
Fighting  
Launching
Question 5: How do objects and processes relate to each other?

Answer: Processes happen to objects.
Question 6: What does a process do when it happens to an object?

Answer: The process transforms the object.
OPM Things (entities, concepts): 

Objects and Processes

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

**Process:** A thing that transforms one or more objects.
OPM’s only two building blocks:

1. **Stateful Object**
2. **Process**

- All the other elements are between things, expressed graphically as **links**.
- Hence, all we need to model the universe are **things** and **relations** among them.
- This is the basis for the **Object-Process Theorem**
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 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
Transforming can be done in three ways:

(1) The process can **consume** the object.

- **Raw Material** → **Manufacturing**
- **Manufacturing** consumes **Raw Material**.
(2) The process can \textit{create} the object

Manufacturing consumes \textit{Raw Material}.
Manufacturing yields \textit{Product}.
The third and last kind of object transformation:

(3) **The process can affect the object.** It does so by changing the object’s state:

*Product can be pre-tested or tested.
Testing changes Product from pre-tested to tested.*
The graphics-text equivalence OPM principle

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

This bimodal representation 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”
Question 9: What are the two major aspects of any system?

- **Structure** – the static aspect: **what** the system is made of.
  - *Time-independent*
- **Behavior** – the dynamic aspect: **how** the system changes over time.
  - *Time-dependent*
- **TIME**  *is the discriminating factor!*
Question 10: What third aspect is specific to human-made systems?

- **Function** – the utilitarian, subjective aspect:
  - Why is the system built?
  - For whom is the system built?
  - Who benefits from operating the system?
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)
System Typology: From Informal to Formal model

Definition of a system: A group of inter-related, possibly interacting parts combined in a way that creates one or more emergent property or capabilities not possessed by the separate parts.

- **Real System**: Systemic and physical.
- **Conceptual System**: Systemic and informatical.

**OPM model**

**OPD – Object-Process Diagram**

**OPL – Object-Process Language**

Real System and Conceptual System are **Systems**. System and Conceptual System are **systemic** and **informatical**. Real System is **systemic** and **physical**.

Recognized and Abstracted Systems

OPD – Object-Process Diagram

OPL – Object-Process Language

Recognized System Unfolded

Artificial System and Naturally Occurring System are Recognized Systems.
Hybrid System is an Artificial System and a Naturally Occurring System.

Stakeholder Group benefits from Utility of Function of Artificial System. Stakeholder Group suffers from Damage of Unintentionally Modified System.
The complete System Typology Model
The corresponding OPL

System exhibits Structure, Behavior, and Architecture.
  Structure is what the system is.
  Behavior is what the system does.
  Architecture consists of Structure and Behavior.
  Architecture enables Function.
Model represents Conceptual System.
Stakeholder Group is physical.
Stakeholder Group suffers from Damage.
Stakeholder Group benefits from Utility.
Environment is environmental and physical.
Real System is physical.
Real System is a System.
Real System and Environment are interact.
Conceptual System is a System.
Recognized System is physical.
Recognized System is a Real System.
Recognized System corresponds to Abstracted System.
Unrecognized System is physical.
Unrecognized System is a Real System.
Abstracted System is a Conceptual System.
Abstracted System models Recognized System.
Abstract System is a Conceptual System.
Naturally Occurring System is physical.
Naturally Occurring System is a Recognized System.
Artificial System is physical.
Artificial System is a Recognized System.
Artificial System exhibits Function.
  Function is what value the system provides.
  Function exhibits Utility.
Hybrid System is physical.
Hybrid System is a Naturally Occurring System and an Artificial System.
Intentionally Modified System is physical.
Intentionally Modified System is a Hybrid System.
Unintentionally Modified System is physical.
Unintentionally Modified System is a Hybrid System.
Unintentionally Modified System exhibits Damage.
Mental Model is a Model.
Shared Model is a Model.
Informal Model is a Shared Model.
Formal Model is a Shared Model.
Conceptual Model is a Formal Model.
Computational Model is a Formal Model.
Hybrid Formal Model is a Conceptual Model and a Computational Model.
OPM ISO 19450:2015 Annex C

Contains a reflective model of OPM (model of OPM in OPM).

C.2 OPM model structure

 Specifies the syntax and semantics of OPM in OPM based on the Universal Ontology of stateful objects and processes that transform them.

1/31/18 Dov Dori © 2018
OPM Resources


- Standard ISO/PAS 19450 OPM

- Website: Enterprise Systems Modeling Laboratory contains
  - journal & conference papers,
  - free OPCAT software, upcoming OPCloud
  - presentations,
  - Projects
  - and more.

- OPCloud: [https://www.opcloud.tech](https://www.opcloud.tech)
Thank you for attending!

dori@technion.ac.il
dori@mit.edu

Visit our **Lab site**; Download OPCAT free

http://esml.iem.technion.ac.il/

Experience **OPCloud**, Cloud-based OPM modeling:

https://www.opcloud.tech/

Join the growing OPM community [Here](https://www.jiscmail.ac.uk/cgi-bin/webadmin?SUBED1=OPM&A=1)
Two Complementary Proofs for Object-Process Theorem:

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

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

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
“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
Iron Dome – A medium-range missile defense system

To be presented
Sample of engineering domains in which OPM has been used

- **Complex, Interconnected, Large-Scale Socio-Technical Systems.** *Systems Engineering* 14(3), 2011.


- **Project-Product Lifecycle Management.** *Systems Engineering*, 16 (4), pp. 413-426, 2013.


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

- All the OPDs, at any detail level, are self-similar.
- They contain only stateful objects, processes, and relations.
## SysML and OPM – a brief comparison

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

1. Model requirement with customer (or Marketing as proxy)
   • This is the problem domain model

2. Specify alternative selection criteria based on value as benefit as cost
   • This will enable decision making once we have alternative concepts
Homeowner is environmental and physical. Homeowner handles Preparing.
Dish Set is environmental and physical. Dish Set can be soiled not loaded or soiled loaded. soiled not loaded is initial.
Dishwasher is physical. Dishwasher consists of Control Subsystem, Door Subsystem, and Rack Set.
Control Subsystem is physical. Door Subsystem is physical.
Door Subsystem exhibits Door Position. Door Position can be latched or unlatched.
latched is initial. unlatched is final.
Rack Set is physical.
Chemistry is environmental and physical.
Chemistry can be internal or inserted.
external is initial.
Location can be unloaded or loaded.
unloaded is initial.
loaded is final.
Preparing is physical.
Preparing exhibits Selected Cycle and Selected Option Subset.
Preparing consists of Dish Loading, Chemisty Filling, Cycle & Option Selecting, Door Closing, and Door Opening.
Preparing zooms into Door Opening, Dish Loading, Chemisty Filling, Cycle & Option Selecting, and Door Closing, as well as Selected Option Subset and Selected Cycle.
Selected Option Subset depends on Selected Cycle.
Selected Cycle can be normal, light, shr, ptopswan, or smartWash.
Door Opening changes Door Position from latched to unlatched.
Dish Loading is physical.
Dish Loading requires Rack Set.
Dish Loading affects Door Subsystem.
Dish Loading changes Location from unloaded to loaded and Dish Set from soiled not loaded to soiled loaded.
Chemisty Filling is physical.
Chemisty Filling requires Door Subsystem.
Summary: OPM Aspect Unification

The three system aspects

- **Function** *(why the system is built)*,
- **Structure** *(static aspect: *what* is the system made of), and*
- **Behavior** *(dynamic aspect: *how* the system changes over time)*

- Are expressed bi-modally, in graphics and equivalent text
- In a **single** model
Question 11: What is the Value which the beneficiary seeks?

- **VALUE IS BENEFIT AT COST**

- **VALUE = BENEFIT – COST**

- **Benefit** comes from function (processes). We want to *maximize* it

- **Cost** comes from form (hardware, stuff, objects). We want to *minimize* it
Question 12: What is System Architecting & Architecture?

- System architecting is mapping *form* to *function* to maximize value to the system’s beneficiary.
- System architecture is the combination of *structure* (*objects*) and *behavior* (*processes*) that maximize the value to the beneficiary.
Summary: OPM Aspect Unification

The three system aspects:

- **Function** (*why* the system is built),
- **Structure** (static aspect: *what* is the system made of), and
- **Behavior** (dynamic aspect: *how* the system changes over time)

- Are expressed bi-modally, in graphics and equivalent text
- In a **single** model