The Maturation of Model-Based Systems Engineering: OPM as the ISO Conceptual Modeling Language Standard
Multiple engineering professionals talk different languages

What language do systems engineers talk?
Required:

A graphical formal language for conceptually conveying system architectures & designs
Systems Engineers Do Have Languages

– Systems Modeling Language – SysML
  • OMG Standard since 2007

– Object-Process Methodology – OPM
  • ISO Standard in the making since 2009
  • OPM book published in 2002
Why Conceptual Modeling?

- Construct a mental picture of the system
- Anchor one’s understanding in visual & textual formalisms for dual channel processing
- Design systems using non-verbal means
- Convert tacit knowledge to explicit knowledge
- Communicate concepts to others
Ontology

A set of concepts for describing a domain (industry, banking, military, healthcare...) and systems within it

Universal Ontology

A set of concepts for describing the universe and systems within it
Towards ontological grounding of model-based systems engineering

Let us try to determine the minimal set of concepts required to model the universe and systems in it

We begin with a series of Socratic questions
First fundamental question:

What are the things that exist in the universe?

Answer:

Objects exist (or might exist)
Second fundamental question:

What are the things that happen in the universe?

Answer:

Processes happen (or might happen)

Processes cannot happen in vacuum!
A Follow-up question:
What are the things to which processes *happen*?

Answer:
Processes *happen* to objects
So:

What do processes do to objects?

Answer:

Processes *transform* objects
What does it mean for a process to transform an object?

Transforming of an object by a process means

1. **creating** (generating) an object
2. **destroying** (consuming) an object
3. **affecting** an object
What does it mean for a process to affect an object?

– A process affects an object by changing its state

– Hence, objects must be stateful; they must have states
Another key question: What are the two complementary aspects from which any system can be viewed?

Answer:

1. **Structure** – the static aspect:
   
   *what* is the system made of?

2. **Behavior** – the dynamic aspect:

   *how* does the system change over time?
What additional aspect pertains to man-made systems?

*Function* – the utilitarian, subjective aspect:

*why* is the system built? for *whom*? *who* benefits from operating it?
The idea behind conceptual modeling

Using graphical symbols, the model expresses real things – **objects** and **processes** – and relations among them.
The Object-Process Theorem

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

Corollary

Using stateful objects, processes, and relations among them, one can model systems in various domains and at any level of complexity.
Proof: Part 1 - necessity

Stateful objects and processes are necessary to specify the two system aspects:

- Specifying the structural, static system aspect requires stateful objects and relations among them
- Specifying the procedural, dynamic system aspect requires processes and relations between them and the objects they transform
Proof: Part 2 - sufficiency

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

- Anything that **exists** can be specified in terms of **stateful objects** and relations among them.
- Anything that **happens** to an object can be specified in terms of **processes** and relations between them and the object they transform.

Q.E.D.
Keys to good conceptual modeling:

1. Telling **processes** apart from **objects**
2. Modeling them **concurrently** to express the interdependence of the systems’ **structure** and **behavior**
3. Managing complexity via **abstraction-refinement**
4. Utilizing dual channel processing: **graphics** and **text**
The Six Leading MBSE Methodologies
(INCOSE Task Force, Estefan, 2008 p 43)

- IBM Telelogic Harmony-SE
- INCOSE Object-Oriented Systems Engineering Method (OOSEM)
- IBM Rational Unified Process for Systems Engineering (RUP SE) for Model-Driven Systems Development (MDSD)
- Vitech Model-Based System Engineering (MBSE) Methodology
- JPL State Analysis (SA)
- Object-Process Methodology (OPM): 2014 – expected ISO 19450 PAS

SysML was not surveyed since it is a language, not a methodology
Object-Process Methodology (OPM)

Things: **Objects** and **Processes**

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

**Process**
A thing that transforms one or more objects
Processes transform objects by

(1) Consuming them:

Manufacturing consumes Raw Material.
Processes transform objects by

(2) Creating them:

Processes transform objects by

(3) Changing their state:

Product can be pre-tested or tested. 
Testing changes Product from pre-tested to tested.
So the OPM Things are:

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

All the rest are relations between things!
Specifying systems with OPM:

The Baggage Handling System
Prepared using OPCAT – [Free download at esml.iem.technion.ac.il]

Object-Process Language (OPL)

Object-Process Diagram (OPD)

Beneficiary

Operand

Attribute

System's function

Attribute values

Passenger

Baggage Handling

Airport

Airline

Baggage

Baggage Location

origin

destination

Baggage exhibits Baggage Location.

Baggage Location can be origin or destination.

origin is initial.

destination is final.

Airport is physical.

Airline is physical.

Baggage Handling is physical.

Baggage Handling requires Airline and Airport.

Baggage Handling changes Baggage Location from origin to destination.
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
Zooming into Baggage Handling

Time line: from the process ellipse top to its bottom
The universality of the object-process ontology: A complementary empirical proof

Q: What do baggage handling and molecular biology have in common?

A: Both can be conceptually modeled with stateful objects and processes that transform them.
An organism is a highly complex system

ORGANS

MOLECULAR BIOLOGY:

CELL

are made of cells

BIOLOGICAL REACTIONS

BIOLOGICAL PATHWAYS

GLYCOLYSIS
Abstract

We propose a Conceptual Model-based Systems Biology framework for qualitative modeling, executing, and eliciting knowledge gaps in molecular biology systems. The framework is an adaptation of Object-Process Methodology (OPM), a graphical and textual executable modeling language. OPM enables concurrent representation of the system’s structure—the objects that comprise the system, and behavior—how processes transform objects over time. Applying a top-down approach of recursively zooming into processes, we model a case in point—the mRNA transcription cycle. Starting with this high level cell function, we model increasingly detailed processes along with participating objects. Our modeling approach is capable of modeling molecular processes such as complex formation, localization and trafficking, molecular binding, enzymatic stimulation, and environmental intervention. At the lowest level, similar to the Gene Ontology, all biological processes boil down to three basic molecular
Stage 1: Verifying that the *in silico* computer simulations are compatible with the experimental results.

The *in silico* computational model represents the experimental knowledge.
Adding a new biological conjecture

- Conjecture evaluation
- Finding knowledge gaps
- Designing and doing new wet lab experiments
Milestones in OPM as the upcoming ISO 19450

- **Paris, France, April 23-24, 2009**: ISO Study Group to explore OPM for modeling standards established, based on RESOLUTION 611 (PARIS 21) – OBJECT PROCESS METHODOLOGY
- **Tokyo, Japan, March 26, 2010**: Normative Draft International Standard preparation decision
- **North Redington Beach, FL, USA, May 12, 2011**: OPM Publically Available Specification and DIS for Model-based standards authoring decision.
- **Haifa, Israel, May 5-7, 2012**
- **Frankfurt, Germany, May 12-15, 2013**
- **Beijing, PRC, May 2014**: RESOLUTION 724 (BEIJING 3) – OPM PD-PAS SUBMISSION
# SysML & OPM Key Features

<table>
<thead>
<tr>
<th></th>
<th>SysML</th>
<th>OPM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of diagram kinds</strong></td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td><strong>Modalities</strong></td>
<td>graphics</td>
<td>graphics &amp; text</td>
</tr>
<tr>
<td><strong>Theoretical foundation</strong></td>
<td>UML</td>
<td>minimal universal ontology</td>
</tr>
</tbody>
</table>
OPM-to-SysML views generation

• We developed an algorithm and software for implementing OPM-to-SysML views generation

• We evaluated the results through an experiment to test the quality of the auto-generated diagrams and their impact on system comprehension by 78 students

• Results: the addition of certain auto-generated SysML views to an OPM system model has increased students’ system comprehension.

OPM to SysML Mapping Challenge

• The mapping is “one-to-many”

• Example – a **Process** in OPM can be mapped in SysML to one of the following:
  – **Use Case** (in a Use Case Diagram)
  – **Operation** of a block (in a Block Definition Diagram)
  – **Action** (in an Activity Diagram)
  – State transition **trigger** or in-state **activity** (in a State Machine Diagram)
  – **Message** (in a Sequence Diagram)
OPM-to-SysML Implementation - Use Case Example

SD1 - ABS Braking in-zoomed
Activity Diagram Example

SD1.2 – Anti Locking in-zoomed

- Signal Set
- Anti Locking
- Signal Converting
- Converted Signal Set
- Signal Processing
- ABS Data
- Wheel Lock Detecting
- Wheel Lock Is Detected: no, yes
- Pulse Set Generating
- Actuating Wheel Signal
- ABS
State Machine Diagram Example

SD1.1 – Order Paying And Supplying in-zoomed
Evaluation

• Experiment done with 78 students
• Examined:
  – Comprehension of a system modeled in OPM, with and without the addition of the automatically generated SysML diagrams
  – Errors and inconsistencies between the original OPM model and the auto-generated SysML views
Experiment Setup

• Two sample systems with two groups of students:

<table>
<thead>
<tr>
<th>System</th>
<th>Group A</th>
<th>Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dishwasher</td>
<td>OPM only</td>
<td>OPM + SysML</td>
</tr>
<tr>
<td>CT scanner</td>
<td>OPM + SysML</td>
<td>OPM only</td>
</tr>
</tbody>
</table>

• Students received models of these systems and were asked to:
  – Answer eight comprehension questions
  – Find errors and inconsistencies among different diagrams of the system model
Evaluation Results

• The addition of the auto-generated SysML diagrams to the OPM model improved the level of comprehension:

<table>
<thead>
<tr>
<th>System</th>
<th>Avg. score OPM only</th>
<th>Avg. score OPM + SysML</th>
<th>t-statistic</th>
<th>p-value (two tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dishwasher</td>
<td>18.97</td>
<td>24.60</td>
<td>-3.763</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>CT</td>
<td>16.50</td>
<td>22.42</td>
<td>-3.617</td>
<td>&lt; 0.001</td>
</tr>
</tbody>
</table>

• No “true positive” errors or inconsistencies were indicated by the students between the OPM and the automatically generated SysML diagrams.
• 74% of the participants (58 of 78) explicitly indicated that the SysML diagrams in addition to the OPM model were helpful in at least one aspect

• SysML diagrams helpfulness in the experiment:

<table>
<thead>
<tr>
<th>SysML Diagram Type</th>
<th>Count</th>
<th>Percentage (of 58)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Definition</td>
<td>27</td>
<td>46.6%</td>
</tr>
<tr>
<td>State Machine</td>
<td>23</td>
<td>39.7%</td>
</tr>
<tr>
<td>Activity</td>
<td>22</td>
<td>37.9%</td>
</tr>
<tr>
<td>Use Case</td>
<td>9</td>
<td>15.5%</td>
</tr>
</tbody>
</table>
Comparison Summary

- OPM and SysML take different approaches in realizing the goal of general-purpose systems conceptual modeling.
- OPM is good for idea generation and rapid prototyping at early architecting and design stages.
- SysML is more appropriate when detailed views are required, during later stages.
- Each language has benefits and drawbacks.
- Neither is by all means “better” than the other.
- There is potential synergy in using both languages.
Summary 1

- Stateful objects, processes that transform them, and relations among them constitute a universal ontology.
- OPM uses this exact ontology.
- This enables modeling systems:
  - In any domain
  - At any level of complexity
  - Using both graphics and text appeals to “both sides of the brain”
• OPM is in the process of becoming ISO standard 19450
• SysML is the OMG standard for systems engineering since 2007
• Using OPM in the early system architecting stages and SysML at later stages can synergistically improve modeling quality and system comprehension
OPM Resources:


- Enterprise Systems Modeling Laboratory website: Papers, software, presentations, projects…
Questions and Answers

Contact: Dov Dori – dori@mit.edu
Thank you for your interest and attention!

Contact: Dov Dori – dori@mit.edu
Sample 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.

- **Model-Based Risk-Oriented Robust Systems Design.** *International Journal of Strategic Engineering Asset Management*, 1(4), pp. 331-354,
Zooming into Origin Baggage Handling

- Airport
  - Origin Airport
  - Airport Personnel
  - Airport Facilities
- Origin Baggage Handling
  - Checking In
  - Security Screening
    - Security Clearance Obtained: no
    - Security Clearance Obtained: yes
  - Sorting & Loading
- Airline Personnel
- Aircraft
- Baggage
- IATA Tag
- Baggage Holder
  - passenger
  - airline
  - security
- Baggage Location
  - aircraft
  - other
  - origin

System design and management
Zooming into Sorting & Loading

Baggage Sorting changes Baggage Holder from security to airline.

OPL sentence example:

```
Baggage Holder security airline
```
Zooming into Decay and Nuclear Import
An example from level 10 OPD:
OPDs are self-similar at any level!
Automation systems and integration — Object-Process Methodology

Systèmes d’automatisation et intégration – Méthodologie du processus-objet

Document type: Publicly Available Specification
Document subtype:
Document stage: (20) Preparatory
Document language: E
STD Version 2.1c2

Expected document size: 190 pages
UML is 1400 pages
SysML is 250 pages
Diagram Type Multiplicity

**SysML**

- Nine diagram types
- Allows detailed and dedicated visual representation focused on a specific aspect
- Semantic overlap (activity, use case, message, action, state transition…)
- Renders the language complicated, increases learning and understanding effort

**OPM**

- One diagram type (OPD) with on-the-fly translation to English (OPL)
- Combines structure and behavior in a single view
- Lacks dedicated diagram types for specific aspects
- But aspects can be highlighted by focusing on objects or processes or states.
Complexity Management in OPM & SysML

• OPM has built-in hierarchy-based abstraction-refinement mechanisms (in-zooming, unfolding…) for managing system complexity

• SysML diagram hierarchies also exist for many of the nine diagram types

• Compared with OPM these hierarchies are
  – less integrated into the language
  – more tool-dependent
  – more complex since it is done separately and differently for various diagram types