Specification and Analysis of Information Systems

Lecture 2

OPM introduction

Dr. Niva Wengrowicz
Object Process Methodology (OPM)
Object Process Methodology (OPM)

- Developed by Prof. Dori from the Technion
- ISO standard -19450
- Holistic graphical and textual methodology to represent complex systems
- Enables representing functional, structural and behavioral aspects of systems using a highly compact set of concepts in a single diagram type and equivalent natural language.
Things and links that connect them are the elements of any system.

Just two types of things:
- Object which can be possibly stateful
- Object with States
- Process

Each thing stands alone as a concept.

Things and states are called entities.

A Link connects two entities.

Ontology? What is ontology?

The nature of being or the kinds of things that have existence (defines the elements/components of the world)
• **Things** and **links** that connect them, are the elements of any system.

• **Just two** types of **things**:
  
  – **Object** which can be possibly stateful – **Object with States**
  
  – **Process**

  – Each **thing** stands alone as a concept.
  
  – **Things** and **states** are called **entities**.

• **A Link** connects two **entities**.

OPM - a Universal Ontology for System Engineering
Object

- An **object** is a **thing** that exists or can exist **physically** or **informatically**/**conceptually**/**logically**

- It can be as **simple** as a block of ice or a data record in a file

- It can be as **complex** as an organization, a human brain, or a galaxy
Object Naming

- Object naming is a capitalized **noun**.
  
  ![Cake](image1)

- It can be a phrases with more than one word
  
  ![Apple Cake](image2)

*note that every word is CAPITALIZED*
Object Naming

• The object **singularity OPM principle**:

  – A name of an object **must be singular**.

  – Convert a plural to singular by adding the word "**Set**“ or “**Group**”

  – "**Set**" and “**Group**” are **OPM reserved words** used for loops and iterations on the set members.

**Ingredients** (e.g., of a Cake) becomes **Ingredient Set**

**Students** (e.g., of this Class) becomes **Student Group**
Object Properties
• A **state** is a **possible situation** at which an **object** can be, or a **value** it can assume, for some positive amount of time.

• A state does **not** stand alone; it has a meaning only within, and in the context of, an object.

• State names are **not capitalized**
Object State

- A **state** is a possible situation at which an **object** can be, or a **value** it can assume, for some positive amount of time.

- A state does **not** stand alone, it has a meaning only within, and in the context of, an object.

- State names are **not capitalized**

**Exercise:** model these examples

- States of the object **Organization** can be **private** or **public**
- States of the object **Record** can be **locked** or **unlocked**

- Lamp
  - off
  - on
Object State

- A **state** is a possible **situation** at which an **object** can be, or a **value** it can assume, for some positive amount of time.

- A state does **not** stand alone. It has a meaning only within, and in the context of, an object.

- State names are **not capitalized**

**Exercise:** model these examples

<table>
<thead>
<tr>
<th>Organization</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>private</td>
<td>unlocked</td>
</tr>
<tr>
<td>public</td>
<td>locked</td>
</tr>
</tbody>
</table>
Object State- initial & final states
Object State - initial & final states

What is wrong?
Object Transformation

- **Object transformation** is the
  - **creation** (generation, construction) of an **object**
  - **consumption** (elimination, destruction) of an **object**
  - **changing** the **state** of an **object**.

- **Transformation** takes a **positive** amount of time.
A **process** is a **thing** that **transforms** an **object**

In other words:

*a process is a pattern of object transformation*

➢ **By definition,**

*a process must be associated with at least one object,*

the one which the **process** **transforms**.

![Diagram of process in Lamp Lighting system](image)
Process

- A **process** is a **thing** that **transforms** an **object**

In other words:

a **process** is a **pattern** of **object transformation**

---

**Exercise:** model these examples

- Freezing is a process that creates an Ice Block
- Melting is a process that destroys an Ice Block
• A process is a thing that transforms an object.

In other words:

A process is a pattern of object transformation.

**Exercise:** model these examples

- Freezing
- Ice Block
- Melting
Process Naming

- The gerund process naming mode ("ing" suffix).

In other words:
A process name is a phrase whose last word should be the gerund form of a verb - a verb with the "ing" suffix.

- If there are several choices, such as in

  Construction VS Constructing
Process Naming

- The **gerund process** naming mode ("ing" suffix).

  In other words:
  A **process** name is a phrase whose **last** word should be the **gerund form** of a verb - a verb with the "ing" suffix.

- If there are several choices, such as in

  Construction VS Constructing

  the latter is preferable.
Process Naming

- The **gerund process** naming mode ("ing" suffix).

  In other words:
  A **process** name is a phrase whose **last** word should be the **gerund form** of a verb - a verb with the "ing" suffix.

- An **object** name can precede the **gerund**.

- Adding an **object** before the **process** qualifies the **process**.

Wall Painting  Roof Painting  Car Painting
Process Naming

- The **gerund process** naming mode ("ing" suffix).

In other words:

A **process** name is a phrase whose **last** word should be the **gerund form** of a verb - a verb with the "ing" suffix.

- The gerund naming convention has two advantages:
  - clarifies the **dynamic nature** of the **process** as a **thing that happens** rather than a **thing that exists**.
  - The **ing** suffix enables **automated detection** of **processes**.
OPM Links

- **Structural links**
  Connection between objects or processes which denote structural relations.

- **Procedural link**
  Connection between an object or its state and a process which describes system's behavior.
Connection between **objects** or **processes** which denote structural relations.

**Registration Form Data** consists of **Name**, **Email Address**, **Password**, **Birth Date**, and **Gender**.

**User Managing** consists of **Self Setting Managing** and **Friend Managing**.
Part II
Change & Effect
Existence & Transformation
Object Transformation

- **Object transformation** is the
  - **creation** (generation, construction) of an **object**
  - **consumption** (elimination, destruction) of an **object**
  - **changing** the **state** of an **object**.

- **Transformation** takes a **positive** amount of time.
Change

- Processes and system dynamics are closely associated with the notion of *change*.

- When we talk about a change in OPM, we need to be specific about what change means.

- A *change* of an object is an alteration in the *state of that object* - a change of an object is reflected in replacing its current state by another state.

- The only thing that can cause this change is a *process*.

- The process causes the change by taking an object at *some state* as input, and outputting it in *another state*. 
Effect

- **Stateful** objects can be **affected** and change.
- This change mechanism underlines the intimate, inseparable link between objects and processes.
- We call this change in state the **effect** of the process on the object.

**Effect** is a change in the state of an object that a process causes.

While the terms "change" and "effect" are almost synonymous, there is a subtle difference in their usage:

- **Effect** refers to what the process does to the object.
- **Change** refers to what happens to the object as a result of the process occurrence.
Existence and Transformation

- **Change** is only one possibility of what can happen to an object when a process acts on it.

- A process affects an object to change it, but it can also do things that are more drastic:
  - It can **generate** an object or **consume** it.
  - The term **transformation** covers the three possible modes by which a process can act on an object: **construction**, **effect**, and **consumption**.

- **Transformation** is the generalization of construction, effect, and consumption, which a process can do when it acts on an object.
Change of State or Change of Identity?

- The extent of the transformation change can vary:
  - **Change in state**
    - If the change is small, we tend to say that the object was altered from one of its states to another while keeping its identity.
  - **Change in object identity**
    - As the extent of the effect grows, so does the difference between the object before the process started and after it ended.
    - At some point, the two are so different, that the human inspector is inclined to think of the object resulting from the process as a newly created object.
    - The object that had existed before the process took place may have been eliminated or at least changed radically.
Transformations in Living Organisms

- In nature, living organisms undergo a striking variety of transformations.
- Some of the transformations are deemed as just a change in state while others are a change in object identity.

**Change in state**
- The transformation from a cub to a grown-up lion is considered a change in the state of a lion from young to adult.
- Similarly, growing of a baby into an adult is considered a change in the person’s state.

**Change in object identity**
- The silkworm, on the other hand, has four distinct forms of existence. It transforms from egg to larva (גּוּז) to pupa (גּוּלָם), to butterfly, which, in turn, lays the eggs of the next silkworm generation.
- Each transformation yields an object that is very distinct from its predecessor in shape and function. The difference is so profound that each such transformation is called metamorphosis.
- From a genetic viewpoint, all are still the same organism.
Transformations in Artificial Objects

- What transformation is "profound" is subjective and context-sensitive.

- Consider, for example, two processes from a manufacturing realm: Molding and Testing.

  **Change in object identity**
  - **Molding** acts on the object **Raw Material** (e.g., plastic), converting it to another object, that we call **Product**.
  - The identity of **Raw Material** changed as a result of the Molding process to the extent that we need to refer to the process outcome by a different name.
  - Hence, the object **Raw Material** has been eliminated or consumed, while a new object, called **Product**, has been created (or constructed, or generated).

  **Change in state**
  - **Testing** only changes product’s state
Procedural Links

- A **procedural link** is a link between a **process** and an **object** or its **state**.

- **Procedural links** provide the **glue that binds and relates objects** to **processes** in an OPD.

- They providing for

  **integration of the system’s structure and behavior** within a **single** model which is one of the most important features of OPM.
Procedural Links

- **Transforming** – A transformee of a process is an object that undergoes a transformation as a result of the occurrence of the process. The transformation can be **construction**, **effect** (change of state) or **consumption**.

- **Enabling** – An enabler of a process is an object that must be present in order for that process to occur, but is **not transformed** as a result of the occurrence of the process.
The three Transformation Links: A File Processing Example

A **construction/result** link is a transformation link that connects a process P with a resultee P.

An **effect** link is a transformation link that connects a process P with an affectee of P.

A **consumption** link is a transformation link that connects a process P with a consumee of P.

Creating yields File.
Deleting consumes File.
Editing affects File.
The three Transformation Links:
A File Processing Example

A **construction/result** link is a transformation link that connects a process P with a resultee P.

An **effect** link is a transformation link that connects a process P with an affectee of P.

A **consumption** link is a transformation link that connects a process P with a consumee of P.
Enabling Links

- Enables are linked to processes through enabling links.

The term **agent** is reserved for a human enabler. An agent can also be a transformee.

An **instrument** of a process is any non-human object. Instrument **does** change, but these changes are either **not significant** enough to be accounted for, or they are out of the system’s scope.

Different structure to OPL’s sentences
OPM Links - Procedural

Construct the OPD based on the OPL sentences

User Signing yields User Data.
User Managing affects User Data.

User Deleting consumes User Data.

User handles User Deleting, User Managing, and User Signing.
OPM Links - Procedural

Construct the OPD based on the OPL sentences

User Signing yields User Data.
User Managing affects User Data.
User Deleting consumes User Data.

User handles User Deleting, User Managing, and User Signing.

Diagram: User Signing, User Managing, User Deleting, User Data, User
Part III

SD
Always start modeling by defining, naming, and depicting the function of the system, which is also its top-level process.
The System Diagram (SD) of the Bread preparing function and the objects involved in it
The System Diagram (SD) of the Operation Room Toolset Handling function and the objects involved in it.
The System Diagram (SD) of the Baggage Handling function and the objects involved in it
The first step in specifying and modeling a system with OPM is to **determine** the **function** of the system.

---

What is the main value of **Dropbox**?

---

What is the main value of **Dropbox**?

---

The **function** is the **main process** of the system.

---

The **function** is designed to **deliver value** to the person or people who gain from using the system.
The first step in specifying and modeling a system with OPM is to **determine** the **function** of the system.

**User Content Availability**
- Computer dependent
- Regardless of computer

**Cloud-Based Files-Hosting**

*User Content Availability can be computer dependent or regardless of computer.*
- Computer dependent is initial.
- Regardless of computer is final.

*Cloud-Based File-Hosting changes User Content Availability from computer dependent to regardless of computer.*
The OPD Top-Down Hierarchy

- SD root diagram (System Diagram) is the most abstract level
- OPDs are hierarchical by construction via recursively refining entities such that each OPD is a refinement of its ancestor:
  - Zooming into processes of interest
  - Expressing object states
  - Unfolding objects or asynchronous processes
  - Deriving dedicated views* for specific aspects

- The “BIG PICTURE” is clear and not lost when looking at details in low-level diagrams
- OPDs should not be too cluttered
- The entire OPD set specifies the system

* With commercial OPCAT version only
In Zooming

Process structure & order
Clarity and Understandability Issues
OPM System Diagram (SD)
Identifying the System’s Beneficiary

- The first step in specifying and modeling a system with OPM is to **determine** the **function** of the system.
  - The function is the **main process** of the system.
  - The function is designed to **deliver value** to the person who gain from using the system.
  - We determine the function of the system by **generalizing** all the customer’s requirements.
  - The function-as-a-seed principle.

Always start modeling by **defining**, **naming**, and **depicting** the **function** of the system, which is also its **top-level process**.
The System Diagram (SD) of Baggage Handling

- Passenger
  - Owns
  - Baggage
- Baggage Handling
  - In-Zooming
  - Baggage Location
    - origin
    - destination
  - Weight
    - owns
After the generalizing customer’s requirements, we can zoom in this main process (“In-Zooming”) and define how to gain this functionality.

Zooming into the **Baggage Handling** function exposes the four sub processes: **Origin Baggage Handling**, **Destination Baggage Handling**, **Baggage Claiming**, and **Lost & Found Baggage Handling**.
The System Map: All the OPDs in one View
**The timeline OPM principle**
*(Time related events also known as Implicit Invocation links)*

- Process Execution Order:
  - the timeline within an in-zoomed process is directed by default from the top of the in-zoomed process ellipse to its bottom.
  - Each process invokes the process(es) immediately below it.

![Diagram of process execution order](image)

Product Terminating zooms into

Product Finishing and

Product Shipping, in that sequence.
The timeline OPM principle
(Time related events also known as **Implicit Invocation links**)

- **Process Execution Order:**
  - the **top-most point** of the process ellipse serves as a **reference point**
  - a process whose reference point is **higher** than its peer **starts earlier**
  - If the reference points of two or more processes are at **the same height** (within some tolerance), these processes start **simultaneously**.

![Diagram](image)

**Processing** zooms into parallel A and B.

**Processing** zooms into A and parallel B and C, in that sequence.
Expressed-Suppressed States

Clarity and Understandability Issues
Expressed-Suppressed States
Expressed-Suppressed States

- Whashing Machine
- Washing Machine State
- Off
- Idle
- Machine Operation Turning
- OPD Object Properties
- General
- Details
- States
- Roles
- Misc.
- Instances
- Add
- Remove
- Props
- View: Implicit
- Autoarrange

- off
- idle
- soak
- rinse
- drain
- spin
Construction Affection and Consumption with Expressed-Suppressed States

**All states expressed**, links touch states

**Non-existent state suppressed**, links touch objects.

**All states suppressed**
Construction Affection and Consumption with Expressed-Suppressed States

All states expressed, links touch states

Non-existent state suppressed, links touch objects.

All states suppressed
Procedural links and Priorities
Procedural links and Priorities

• Zoom-out Process P
Procedural links and Priorities

• Zoom-out Process P
# Procedural links and Priorities

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Priority</th>
<th>Symbol</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transforming</td>
<td>Consumption</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Result</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Input</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Effect</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enabling</td>
<td>Agent</td>
<td>4</td>
<td>-●</td>
<td>state</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td>Instrument</td>
<td>5</td>
<td>-○</td>
<td>state</td>
<td>Object</td>
</tr>
<tr>
<td>Condition</td>
<td>Condition</td>
<td>6</td>
<td>-○</td>
<td>state</td>
<td>Object</td>
</tr>
</tbody>
</table>
Procedural links and Priorities

• Zoom-out Process P
The Distributivity of Procedural Links

A procedural link that is attached to the in-zoomed ellipse of a process is considered to be attached to each one of the subprocesses inside the in-zoomed process.
Installing OPCAT

• Opcat is the OPM modeling software environment we use

Dov Dori Model-Based System Engineering Part I book draft
ISO 19450 OPM Standard Draft January 2014 File
OPM - OPD & OPL

OPD
object-process diagram

OPL
object-process language

new
OPM - OPD & OPL

OPCAT automatically generates the OPL sentences.

OPD object-process diagram

OPL object-process language