Object–Process Methodology

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

INTRODUCTION

Capturing the knowledge about existing systems and analysis and design of conceived systems requires an adequate methodology, which should be both formal and intuitive. Formality is required to maintain a coherent representation of the system under study, while the requirement that the methodology be intuitive stems from the fact that humans are the ultimate consumers of the knowledge. Object-Process Methodology (OPM) is a vehicle for knowledge representation and management that perfectly meets the formality and intuition requirements through a unique combination of graphics and natural language.

Function, structure, and behavior are the three main aspects that systems exhibit. Function is the top-level utility that the system provides its beneficiaries who use it or are affected by it, either directly or indirectly. The system's function is enabled by its architecture—the combination of structure and behavior. The system's architecture is what enables it to function so as to benefit its users.

Most interesting, useful, and challenging systems are those in which structure and behavior are highly intertwined and hard to separate. For example, in a manufacturing system, the manufacturing process cannot be contemplated in isolation from its inputs—raw materials, model, machines, and operators—and its outputs—the resulting product. The inputs and the output are objects, some of which are transformed by the manufacturing process, while others just enable it.

Modeling of complex systems should conveniently combine structure and behavior in a single model. Motivated by this observation, OPM (Dori, 1995, 2002) is a comprehensive, holistic approach to modeling, study, development, engineering, evolution, and lifecycle support of systems. Employing a combination of graphics and a subset of English, the OPM paradigm integrates the object-oriented, process-oriented, and state transition approaches into a single frame of reference. Structure and behavior coexist in the same OPM model without highlighting one at the expense of suppressing the other to enhance the comprehension of the system as a whole.

Rather than requiring that the modeler views each of the system's aspects in isolation and struggle to mentally integrate the various views, OPM offers an approach that is orthogonal to customary practices. According to this approach, various system aspects can be inspected in tandem for better comprehension. Complexity is managed via the ability to create and navigate across possibly multiple detail levels, which are generated and traversed through by several abstraction/refinement mechanisms.

Due to its structure-behavior integration, OPM provides a solid basis for representing and managing knowledge about complex systems, regardless of their domain. This chapter provides an overview of OPM, its ontology, semantics, and symbols. It then describes applications of OPM in various domains.

THE OPM ONTOLOGY

The elements of the OPM ontology, shown in Figure 1, are divided into three groups: entities, structural relations, and procedural links.

Entities

Entities, the basic building blocks of any system modeled in OPM, are of three types: stateful objects, namely *objects* with *states*, and *processes*. As defined below, processes transform objects by (1) creating them, (2) destroying them, or (3) changing their state. The symbols for these three entities are respectively shown as the first group of symbols at the left-hand side of Figure 1, which are the symbols in the toolset available as part of the GUI of OPCAT 2 (Dori, Reinhartz-Berger et al., 2003).

Figure 1. The three groups of OPM symbols in the toolset of OPCAT 2

OPM Things: Objects and Processes

Objects are (physical or informatical) things that exist, while *processes* are things that transform (create, destroy, or change the state of) objects. Following is a set of basic definitions that build upon each other.

An object is a thing that exists.

Objects are the things that are being transformed in the system.

Transformation is generation (creation) or consumption (destruction) of an object, or a change of its state.

Processes are the things that transform objects in the system.

A process is a thing that represents a pattern of object transformation.

Table 1 shows the OPM things and their basic attributes. The third column on Table 1 contains a description of each thing or attribute and below it the syntax of the corresponding sentence in Object-Process Language (OPL)—a subset of English that reflects the graphical representation. In OPL, bold Arial font denotes non-reserved phrases, while non-bold Arial font denotes reserved phrases. In OPCAT, various OPM elements are colored with the same color as their graphic counterparts (by default, objects are green, processes are blue, and states are brown).

Objects and processes are collectively called *things*.

The first two lines of Table 1 show the symbol and a description of the two types of OPM things. The next two lines show two basic attributes that things can have: essence and affiliation.

Essence is an attribute that determines whether the thing is physical or informational.

The default essence is informatical. A thing whose essence is physical is symbolized by a shaded shape.

Affiliation is an attribute that determines whether the thing is environmental (external to the system) or systemic.

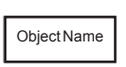
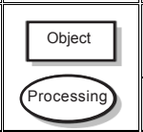
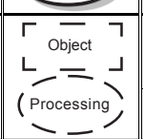
The default affiliation is systemic. A thing whose affiliation is environmental is symbolized by a dashed contour.

OPM States

Objects can be stateful, that is, they may have one or more states.

A state is a situation at which an object can exist at certain points during its lifetime or a value it can assume.

Table 1. Things of the OPM ontology and their basic attributes

Thing / Attribute	Symbol	Description / OPL sentence
Object		A thing (entity) that has the potential of stable, unconditional physical or mental existence.
		Object Name is an object.
Process		A thing representing a pattern of transformation that objects undergo.
		Processing is a process.
Essence		An attribute that determines whether the thing (object or process) is physical (shaded) or informational.
		Processing is physical.
Affiliation		An attribute that determines whether the thing is environmental (external to the system, dashed contour) or systemic.
		Processing is environmental.