

# OPCAT — Object–Process Case Tool: an Integrated System Engineering Environment (ISEE)

Dov Dori, Arnon Sturm  
*Demonstrated by Arnon Sturm*

The William Davidson Faculty of Industrial Engineering and Management  
Technion, Israel Institute of Technology, Haifa 32000, Israel  
dori@ie.technion.ac.il, sturm@tx.technion.ac.il

**Abstract.** This demonstration concerns system development methodologies and their supporting CASE products. The Object–Process Methodology (OPM) integrates system structure and behavior within one model and manages complexity through a scaling mechanism that controls the visibility of things in the system. The demonstration presented OPM principles and its application through OPCAT — Object–Process CASE Tool — the product that supports OPM.

## Background

Object–Process Methodology (OPM) is a system development approach that integrates structure and behavior of the system within a single unifying model. The conventional wisdom has been that there is an inherent dichotomy between object– and process–oriented approaches, and that it is not possible to combine these two essential aspects of any system into one coherent integral frame of reference. This misconception has accompanied systems analysis to the extent that even the accepted UML standard (Booch and Rumbaugh, 1995; Booch and Rumbaugh, 1996) maintains the separation between structure and behavior, and spreads analysis activities across no less than eight types of models that use different diagram types.

## What was demonstrated

In the first part of the demonstration, we present an overview of the OPM. Contrary to the accepted view, that structure and behavior cannot be merged, at the heart of the Object–Process Methodology is the fusion of structural and procedural aspects of a system into a single, unifying model.

OPM distinguishes between objects and processes as two types of things that have equal status and importance in the specification of a system. The OPM model shows how objects interact with each other via processes, such that both the structural and the procedural system aspects are adequately represented. The underlying observation of the Object–Process paradigm is that every thing in the universe of interest is either a process or an object. This opens the door for the possibility of modeling a system using a *single* model that faithfully defines and describes both its structure and behavior. These two major aspects of any system are represented without suppressing

one another. Structural relations — primarily aggregation, generalization and characterization — and procedural relations, which model the behavior of the system over time, are seamlessly intertwined to provide a comprehensive understanding of the system.

The Object–Process Diagram (OPD) — the graphic expression of a system or part of it, analyzed by OPM — is a concise and effective visual language. It incorporates elements from both process–oriented approaches (notably DFD and its derivatives) and object–oriented ones. The Object–Process Language (OPL) provides for a textual, natural–language–like equivalent specification of the system specified through the OPD set. OPL is designed to be read as natural English, albeit with stringent and limited syntax, such that no prior knowledge in analysis methodologies is required. OPL serves as both a feedback mechanism to the prospective customer and as the engine for activities that follows the design stage, notably code generation and database scheme generation.

The second part of the demonstration we presented a case study of a corporate foreign travel management system that demonstrates OPM features and OPCAT current functionality. We exemplified how OPDs are constructed and what symbols they consist of. Through OPCAT’s GUI, we demonstrated OPM’s expressive power, including its zooming in/out and unfolding/folding scaling mechanisms. Finally, we demonstrated the translation of OPDs to OPL using the OPL syntax and the equivalence between the alternative graphical and textual representations.

### **Where to find the product**

URL: <<http://iew3.technion.ac.il:8080/~dori/opcathp/index.htm>>

### **References**

- Dori, D., Object–Process Analysis: Maintaining the Balance Between System Structure and Behavior. *Journal of Logic and Computation*, 5, 2, pp. 227–249, 1995
- Dov Dori, Unifying System Structure and Behavior through Object–Process Analysis. *Journal of Object–Oriented Analysis*, July–August, pp. 66–73, 1996
- Dov Dori and Moshe Goodman, On Bridging the Analysis–Design and Structure–Behavior Grand Canyons with Object Paradigms. *Report on Object Analysis and Design*, 2, 5, pp. 25–35, January–February 1996
- Dov Dori and Moshe Goodman, From Object–Process Analysis to Object–Process Design, *Annals of Software Engineering*, Vol. 2, pp. 20–25, 1996
- Doron Meyersdorf and Dov Dori, The R&D Universe and Its Feedback Cycles: an Object–Process Analysis. *R&D Management*, 27, 4, pp. 333–344, October 1997
- Mor Peleg and Dov Dori, Extending the Object–Process Methodology to Handle Real–Time Systems. *Journal of Object–Oriented Programming* (accepted July 1998)