



## **Chapter VII**

# **The Application of FOOM Methodology to IFIP Conference Case Study**

Judith Kabeli  
Ben-Gurion University, Israel

Peretz Shoval  
Ben-Gurion University, Israel

### **ABSTRACT**

*FOOM (Functional and Object-Oriented Methodology) is an integrated methodology for information systems' analysis and design, which combines two essential software-engineering paradigms: the functional/data approach (or process-oriented) and the object-oriented (OO) approach. Having applied FOOM in a variety of domains, this chapter presents the application of the methodology to the specification of the IFIP Conference system. We focus on the analysis and design phases. FOOM-analysis phase includes data modeling and functional analysis activities and produces an initial Class Diagram and a hierarchy of OO data flow diagrams (OO-DFDs). The products of the design phase include: (a) a complete class diagram; (b) object classes for the menus,*

*forms and reports and (c) a behavior schema, which consists of detailed descriptions of the methods and the application transactions, expressed in pseudocode and message diagrams.*

## INTRODUCTION

This chapter provides a brief description of FOOM methodology, along with its application to IFIP Conference case study. A more detailed description of FOOM can be found in Shoval and Kabeli (2001). The description of the IFIP Conference case study is provided in Mathiassen, Munk-Madsen, Axel Nielsen and Stage (2000), who demonstrate the application of their OOA&D to the IFIP case. The objective is to show how the FOOM methodology, which combines the process and object-oriented paradigms, is suitable for analyzing and designing business-oriented information systems.

## BACKGROUND

Many paradigms for system analysis and design have been proposed over the years. Early approaches have advocated the functional approach (DeMarco, 1978; Yourdon & Constantine, 1979). The development of OO programming languages gave rise to a new approach that maintains that in order to develop information systems in such languages, it is recommended to perform OO analysis and design. Many OO methodologies were developed (e.g., Booch, 1991; Coad & Yourdon, 1990, 1991; Jacobson, 1992; Martin & Odell, 1992; Rumbaugh, Blaha, Premerlani, Eddy & Lorensen, 1991; Shlaer & Mellor, 1988, 1992; Wirfs-Brock, Wilkerson & Wiener, 1990), and the area is still evolving. The multiplicity of diagram types in the OO approach has been a major motivation for developing the Unified Modeling Language (UML) (see Booch, Rumbaugh & Jacobson, 1999; Clee & Tepfenhart, 1997; Larman, 1998; Maciaszek, 2001; UML Rose, 1998). UML was developed in order to produce a standard (unified) modeling language. It consists of several types of diagrams with well-defined semantics and syntax, which enables the presentation of a system from different point of views.

Information systems development is a multiphase process in which the analysis and design are of primary importance. Therefore it is vital to examine which approaches and methods are appropriate to perform each of these phases. On the one hand, those who adopt the OO approach claim that using data abstraction at the analysis phase, producing a model of reality by means of classes, is preferable to producing a functional model, because the real world consists of objects. However, as far as we know, no such study has shown that the OO approach is more effective than the functional/data approach in the development of business-oriented information systems.

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/application-foom-methodology-ifip-conference/28112](http://www.igi-global.com/chapter/application-foom-methodology-ifip-conference/28112)

## Related Content

---

### Design Diagrams as Ontological Sources: Ontology Extraction and Utilization for Software Asset Reuse

Kalapriya Kannanand Biplav Srivastava (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1250-1279).

[www.irma-international.org/chapter/design-diagrams-ontological-sources/29445](http://www.irma-international.org/chapter/design-diagrams-ontological-sources/29445)

### Coordinating Enterprise Services and Data: A Framework and Maturity Model

Keith R. Worfolk (2013). *Service-Driven Approaches to Architecture and Enterprise Integration* (pp. 241-274).

[www.irma-international.org/chapter/coordinating-enterprise-services-data/77952](http://www.irma-international.org/chapter/coordinating-enterprise-services-data/77952)

### Hurricane Damage Detection From Satellite Imagery Using Convolutional Neural Networks

Swapandeep Kaur, Sheifali Gupta, Swati Singhand Isha Gupta (2022). *International Journal of Information System Modeling and Design* (pp. 1-15).

[www.irma-international.org/article/hurricane-damage-detection-from-satellite-imagery-using-convolutional-neural-networks/306637](http://www.irma-international.org/article/hurricane-damage-detection-from-satellite-imagery-using-convolutional-neural-networks/306637)

### A Proactive Approach to Intrusion Detection in Cloud Software as a Service

Baldev Singhand Surya Narayan Panda (2015). *Achieving Enterprise Agility through Innovative Software Development* (pp. 287-305).

[www.irma-international.org/chapter/a-proactive-approach-to-intrusion-detection-in-cloud-software-as-a-service/135232](http://www.irma-international.org/chapter/a-proactive-approach-to-intrusion-detection-in-cloud-software-as-a-service/135232)

### Improving Novice Programmers' Skills through Playability and Pattern Discovery: A Descriptive Study of a Game Building Workshop

Thiago Schumacher Barcelos, Roberto Muñoz Sotoand Ismar Frango Silveira (2015). *Human Factors in Software Development and Design* (pp. 141-172).

[www.irma-international.org/chapter/improving-novice-programmers-skills-through-playability-and-pattern-discovery/117300](http://www.irma-international.org/chapter/improving-novice-programmers-skills-through-playability-and-pattern-discovery/117300)