Chapter XIII

Quality of Analysis Specifications: A Comparison of FOOM and OPM Methodologies

Judith Kabeli, Ben-Gurion University, Israel
Peretz Shoval, Ben-Gurion University, Israel

ABSTRACT

Functional and Object Oriented Methodology (FOOM) combines two essential software-engineering paradigms: the functional (process-oriented) approach and the object-oriented (OO) approach. The two main products of FOOM’s analysis phase are two models: a data model in the form of an initial class diagram and a functional model in the form of OO-DFDs (a hierarchy of data flow diagrams including data classes). We evaluate the quality of these models by comparing them with the quality of equivalent analysis models products by Object-Process Methodology (OPM), which also combines the functional and object-oriented approaches, using a unified diagrammatic notation. The comparison is based on a controlled experiment which measured the correctness of the analysis models (specifications) produced by the two methodologies. The results reveal that the quality of models produced by FOOM is better than those produced by OPM.

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
INTRODUCTION

FOOM is a methodology for analysis and design of information systems that combines two essential software-engineering paradigms: the functional approach (or process-oriented) and the object-oriented (OO) approach (Shoval & Kabeli, 2001). This methodology utilizes well-known techniques, such as DFDs (Data Flow Diagrams), and provides simple visual modeling and notations. It covers the structural and behavior aspects of a system through the analysis and design phases and provides a natural and smooth transition from one phase to the other. Object-Processes Methodology (OPM) is another methodology for analysis and design of information systems that combines the process and object-oriented approaches, providing a unified notation for the structural and behavior aspects of a system (Dori, 2002). Since the two methodologies integrate the functional and the object-oriented approaches, we found it interesting to compare them. Methodologies can be evaluated and compared on various dimensions, for example, quality of the analysis or design products, comprehensibility, learn-ability, ease of use, preference by users/developers, and more. In an earlier study, we have compared these methodologies from the point of view of users by measuring comprehension of the analysis models created with these methodologies (Kabeli & Shoval, 2005). Our objective in this follow-up study is to compare FOOM and OPM from the points of view of analysts by measuring the quality of the models they produce.

The next section provides a background on FOOM and OPM methodologies, concentrating on the analysis phase, followed by a section that overviews related studies on experimental evaluation and comparison of methodologies, in particular from the perspective of quality of specifications. The fourth section describes this experiment, the fifth section analyzes the results, and the last section concludes and proposes further research directions.

BACKGROUND ON FOOM AND OPM METHODOLOGIES

Essentials of FOOM Methodology

FOOM combines two essential software engineering paradigms: the functional and the OO approaches (Kabeli & Shoval, 2003a; Shoval & Kabeli, 2001). Although there are fundamental differences between the two approaches, especially in the strategy used for problem decomposition and in life cycle, the combined approach adopted in FOOM takes advantages of both. FOOM breaks down a system into processes, as in the traditional structured analysis approach, and into objects, as in the OO approach.

FOOM uses well-known visual modeling techniques and a small number of diagram types encompassing the analysis and design phases of development,
Related Content

Temporal Interoperability in Multi+Temporal Databases
www.igi-global.com/article/temporal-interoperability-multi-temporal-databases/51189?camid=4v1a

Rewriting and Efficient Computation of Bound Disjunctive Datalog Queries
www.igi-global.com/chapter/rewriting-efficient-computation-bound-disjunctive/11205?camid=4v1a

Current Approaches and Future Trends of Ontology-Driven Geographic Integration
www.igi-global.com/chapter/current-approaches-future-trends-ontology/20733?camid=4v1a

Methods for the Identification of Data Outliers in Interactive SQL
www.igi-global.com/article/methods-identification-data-outliers-interactive/51083?camid=4v1a