Chapter 3
Synthesis of MOF, MDA, PIM, MVC, and BCE Notations and Patterns

Jaroslaw Zelinski
https://orcid.org/0000-0002-8032-4720
Independent Researcher, Poland

ABSTRACT

Publications, including academic handbooks, contain numerous inconsistencies in the descriptions of applications of architectural methods and patterns hidden under the abbreviations such as MOF, MDA, PIM, MVC, BCE. An efficient analysis and the following software design, particularly when we are speaking of projects realized in large teams, requires standardization of the production process and the applied patterns and frameworks. This study attempted to sort out the system of notations describing this process and used to describe architectural patterns. Analysis of key notations—MOF and MDA, patterns MVC and BCE—was carried out, and a consistent system combining them into a whole was created.

INTRODUCTION

The study objective was to verify the current state of design methods and development of a concise system of notations and design patterns in the field of software logic design as its abstract model. Numerous publications on the analysis and design in the field of software engineering use the names of MVC (Model View Controller) and BCE (Boundary Control Entity) design patterns and the PIM (Platform Independent Model) model (OMG MDA, 2016; OMG MOF, 2016). Considering the frequent
often not too minor discrepancies in the interpretation of these methods and patterns, the author made an attempt to order their mutual interactions. This thesis is based on foundations of mentioned notations only. Main audience of my paper are people using formalism described in OMG notations.

BACKGROUND

In this study, Object Management Group notation systems have been used. MOF (Meta Object Facility) specification describes three abstraction levels: M1, M2, M3 and level M0 that is real items (OMG MOF, 2016). M0 is a real system, M1 level is abstraction of the items of this system (its model). Level M2 comprises of relationships between classes of these objects (names of their sets) that is system metamodel. M3 level is a meta-metamodel describing the modeling method with the use of named elements with specified semantics and syntactic.

The analysis and design process is based on the MDA (Model Driven Architecture) specifications. This process has three phases understood as creation of subsequent models: CIM (Computation Independent Model), PIM (Platform Independent Model), PSM (Platform Specific Model) and code creation phase. The CIM model is documented with the use of BPMN (Business Process Model and Notation) (OMG BPMN, 2013) and SBVR notation (Semantic of Business Vocabulary and Rules) (OMG SBVR, 2017). These are, respectively: business process models and notation models and business rules. PIM and PSM models are documented with the use of UML notation (Unified Modeling Language) (OMG UML, 2017).

Between CIM and PIM models, determination of the list of application services (system reactions) occurs, whose realization mechanism is described by PIM model. The standard pattern used for modeling application architecture is MVC pattern. Component Model of this pattern is modeled with the use of the BCE architectural pattern.

Semiotics vs. UML

Semiotics, as a science dealing with symbols and their meanings, provides us with the tool enabling determination of relationships between an object (thing), its name (expression) and definition of notation represented by the name (or sign, meaning). These relationships are referred to as the semiotic triangle. Figure 1 represents this triangle on the left (OMG SBVR, 2017).

The UML notation (OMG UML, 2017) operates instance classifier and class notations. To the right, Figure 1 demonstrates an equivalent to semiotic triangle expressed with those terms.
Related Content

Construction of Shadow Model by Robust Features to Illumination Changes
[www.igi-global.com/article/construction-of-shadow-model-by-robust-features-to-illumination-changes/105631?camid=4v1a](www.igi-global.com/article/construction-of-shadow-model-by-robust-features-to-illumination-changes/105631?camid=4v1a)

Runtime Safety Assurance for Adaptive Cyber-Physical Systems: ConSerts M and Ontology-Based Runtime Reconfiguration Applied to an Automotive Case Study
[www.igi-global.com/chapter/runtime-safety-assurance-for-adaptive-cyber-physical-systems/186905?camid=4v1a](www.igi-global.com/chapter/runtime-safety-assurance-for-adaptive-cyber-physical-systems/186905?camid=4v1a)
Execution Management for Mobile Service-Oriented Environments
www.igi-global.com/chapter/execution-management-mobile-service-oriented/66793?camid=4v1a

Integrating DSLs into a Software Engineering Process: Application to Collaborative Construction of Telecom Services
www.igi-global.com/chapter/integrating-dsls-into-software-engineering/71828?camid=4v1a