Chapter IX

Integrating Fact-oriented Modeling with Object-oriented Modeling

Terry Halpin
Microsoft Corporation, USA

ABSTRACT
Although the Unified Modeling Language (UML) provides deep support for the design of object-oriented applications, it currently lacks a number of features that have proved very valuable for conceptual analysis, especially for modeling of database applications. One promising solution to this problem is to use a fact-oriented approach such as Object Role Modeling (ORM) for the initial conceptual analysis and then transform this to UML. This paper provides a comparative evaluation of the two approaches, specifically for data modeling and shows how to get the best out of both by using them in combination. Apart from its advantages for conceptual analysis, ORM’s richer and more rigorous constraint specification mechanisms can help to address gaps and inconsistencies in the current UML specification.

INTRODUCTION
Since its acceptance by the Object Management Group (OMG) as a language for object-oriented (OO) analysis and design, the Unified Modeling Language (UML) has become very popular for designing OO program code. It is especially suited for this purpose, covering both data and behavioral aspects, and allowing OO-
implementation details to be directly expressed (e.g., attribute visibility and direc-
tional navigation across associations). Although not yet widely used for designing
database applications, UML can be used for this task also, since its class diagrams
effectively provide an extended Entity-Relationship (ER) notation that can be
annotated with database constructs (e.g., key declarations).

Although the UML specification (OMG, 1999) promotes a use-case driven,
iterative approach to modeling, it has little else to say about the modeling process
itself. While various proprietary methods (e.g., Jacobson et al., 1999) have been
proposed to flesh out this process, they differ in details and offer little support for the
conceptual analysis of information. In practice, some very flawed UML data models
are being used for database design. We believe these defects can be avoided by using
fact-oriented modeling as a precursor to object-oriented modeling in UML. Object-
Role Modeling (ORM) is the main exemplar of the fact-oriented approach, and
though less well-known than UML, is being used productively in many countries
and is supported by various CASE tools. The use of ORM’s analysis techniques
and notations to augment UML for modeling information is the main concern of
this chapter.

The rest of this chapter is structured as follows. Section 2 provides a compara-
tive overview of UML and ORM, based on linguistic design criteria. Section 3
focuses on “data use cases”, illustrating their role in guiding the modeling process
as a joint activity between modeler and subject matter expert. Section 4 illustrates
how ORM’s rigorous semantics can compensate for some deficiencies in the UML
specification. Section 5 discusses transformations between the two approaches,
enabling modelers to get the benefits of both. Section 6 concludes by summarizing the
main ideas, identifying areas of future research, and listing references for further reading.

A Brief Comparison of UML and ORM

The UML notation includes a vast number of symbols from which various
diagrams may be constructed to model different perspectives of an application. The
main diagram types are Use Case diagrams, Static Structure diagrams (Class
diagram, Object diagram), Behavior diagrams (Statechart, Activity diagram), Inter-
action diagrams (Sequence diagram, Collaboration diagram), and Implementation
diagrams (Component diagram, Deployment diagram). Since this paper focuses on
conceptual data modeling, we restrict our discussion of UML to its class and object
diagrams, including supplementary textual annotations. The reader unfamiliar with
the syntax and semantics of UML may refer to its specification (OMG, 1999), a
simple introduction (Fowler, 1997) or a detailed treatment (Booch et al., 1999;
Rumbaugh et al., 1999). An in-depth discussion of UML for database design (using
a slightly different notation) may be found in Blaha and Premerlani (1998). At the
time of writing, the latest specification of UML was version 1.3. Some very minor
modifications are expected for version 1.4, leaving any substantial changes for
version 2.0 early in the 21st century.

Object-Role Modeling is a conceptual modeling method, so-called because it
views the world as a set of objects (entities or values) that play roles (parts in
relationships, which may be unary, binary or longer). For example, you are now
playing the role of being awake (a unary relationship involving just you) and also the
Building a Paperless Service: Making the Internship Connection
Theresa M. Vitolo and Aaron J. Sparks (1999). *Success and Pitfalls of Information Technology Management* (pp. 120-131).
www.igi-global.com/article/building-paperless-service/33485?camid=4v1a

ERP Lifecycle: A Retirement Case Study
www.igi-global.com/article/erp-lifecycle-retirement-case-study/73790?camid=4v1a