IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITB8125

Chapter V

Architecture and Implementation Issues

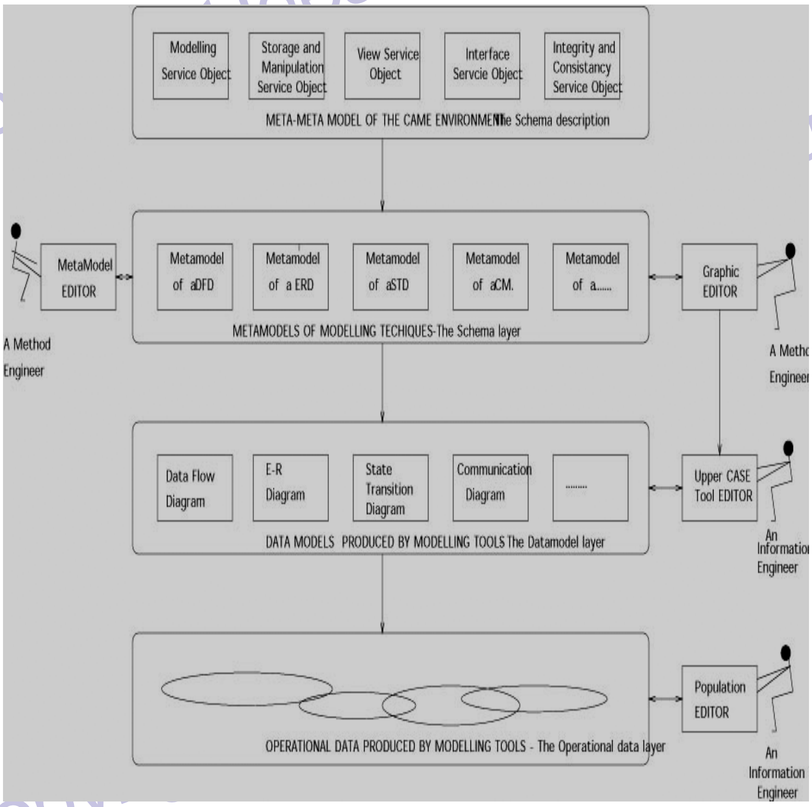
Ajantha Dahanayake
Delft University of Technology, The Netherlands

Historically the focus is on the theory of how problem-specific systems design tools can be supported by a Computer Aided Method Engineering (CAME) environment based on service object representation. To arrive at an implementation model, the conceptual model of the service object representation must be formalized. This theory is feasible when there is adequate computer support. Many researchers have emphasized strongly that requirement specification languages should have a rigorous formal basis; however, this need for formality has not been generally acknowledged in the field of information systems development. Most organizations and research groups tend to define their own methods using techniques advocated within such methods that often have no formal foundation. Discussions of modeling techniques are based on numerous examples, mostly using diagrams and notational conventions, to provide a popular style for the definition of new concepts and their behavior. In a CAME environment however, which gives the freedom to specify a modeling technique from scratch, it is difficult to avoid deficiencies such as inconsistency, lack of structure, over specification, incompleteness, ambiguity, and redundancy without using a formal approach. In automated support a formal model is used to provide stable specifications for implementation. In fact, an implementation can be seen as another, enormously detailed formal description, usually in an imperative programming language. To implement this sophisticated automated support, formal specifications of the CAME service description with adequate formal reasoning were derived earlier.

In this chapter the concentration is on using representation formalism to construct a problem-specific CAME environment. Such an automated support environment must be provided for the information systems design stage in particular for the required UpperCASE tools according to the methods chosen for the problem situations. The vision is that CAME environments must function as a service-based, object-oriented MetaCASE environment that offers the services required for modeling tools, and using a mechanism to interpret the required modeling knowledge and changing the visual representation to the required form using a graphic object binding mechanism. Further, this environment must offer a mechanism for the populations of models specified according to such UpperCASE tools.

According to the service description, a CAME environment consists of five major services. Figure 1 provides a general architecture of a service object based CAME environment that is able to support the activities of users. Two types of users can be identified: one, the 'method engineers' that apply a *meta model editor* to specify meta models of design tools according to problem specific design activities.

Figure 1: The general architecture of a service object based CAME environment


Modelling Service Object
Storage and Manipulation Service Object
View Service Object
Interface Servcie Object
Integrity and Consistancy Service Object
META-META MODEL OF THE CAME ENVIRONMENT The Schema description
MetaModel EDITOR
A Method Engineer
Metamodel of aDFD
Metamodel of a ERD
Metamodel of aSTD
Metamodel of aCM.
Metamodel of a......
METAMODELS OF MODELLING TECHIQUES-The Schema layer
Graphic EDITOR
A Metho
Engineer
Data Flow Diagram
E-R Diagram
State Transition Diagram
Communication Diagram
....
DATA MODELS PRODUCED BY MODELLING TOOLS The Datamodel layer
Upper CASE Tool EDITOR
An Informatio
Engineer
OPERATIONAL DATA PRODUCED BY MODELLING TOOLS - The Operational data layer
Population EDITOR
An Information Engineer