Distributed Model Management: Current Status and Future Directions

Omar F. El-Gayar
Dakota State University, USA

Amit V. Deokar
Dakota State University, USA

INTRODUCTION

Modern organizations are faced with numerous information management challenges in an increasingly complex and dynamic environment. Vast amounts of data and myriads of models of reality are routinely used to predict key outcomes. *Decision support systems* (DSS) play a key role in facilitating decision making through management of quantitative models, data, and interactive interfaces (Power, 2000). The basic thrust of such applications is to enable decision-makers to focus on making decisions rather than being heavily involved in gathering data and conceiving and selecting analytical decision models. Accordingly, the number and complexity of decision models and of modeling platforms has dramatically increased, rendering such models a corporate (and national) resource (Muhanna & Pick, 1994).

Further, Internet technology has brought many new opportunities to conduct business electronically, leading to increased globalization. Managers and decision makers are increasingly collaborating in distributed environments in order to make efficient and effective use of organizational resources. Thus, the need for distributed decision support in general, and model sharing and reuse in particular, is greater today than ever before. This has attracted significant attention from researchers in information systems-related areas to develop a computing infrastructure to assist such distributed model management (Krishnan & Chari, 2000).

In this article, we focus on distributed model management advances, and the discussion is organized as follows. The next section provides a background on model management systems from a life-cycle perspective. This is followed by a critical review of current research status on distributed decision support systems from a model management viewpoint with a particular emphasis on Web services. Future trends in this area are then discussed, followed by concluding remarks.

BACKGROUND

The term model has been reviewed by many researchers and some common elements that characterize models have been noted. Krishnan and Chari (2000) depict a *model* (or a *model schema*) as a formal abstract representation of a decision problem. In other words, models can be conceived as specific formulations of decision situations amenable to certain problem solving techniques, such as simple linear regression, or a linear programming (LP) product mix formulation (Chang, Holsapple, & Whinston, 1993). Examples of models include a demand forecasting model for predicting customer calls in a call center and a production planning model to decide optimal product quantities to be produced. *Model instances* represent specific decision making situations created by instantiating model schemas with appropriate data, and are amenable to computational execution using *model solvers* to determine *model solutions*. Sometimes, models are generically considered as computational units (Orman, 1998) or objects (Lenard, 1993).

In general, models can be seen to conform to a *modeling lifecycle*, consisting of a complex, iterative process during which several modeling tasks need to be accomplished (Krishnan & Chari, 2000) (see Figure 1). Some of the modeling tasks are computationally intensive, while others are more subjective and need human judgment and domain expertise. Supporting the modeling life-cycle encompasses a variety of functionalities. For example, model creation may involve description, formulation, selection, integration, composition, and reuse of models. While model formulation focuses on the knowledge elicitation involved in the development of new models, model selection, composition, and integration aim at leveraging existing repositories of existing models. Model implementation is concerned with issues related to creating model representation amenable to execution by solvers. Issues of model-data, model-solver, and model-paradigm independence are of critical importance. Post-solution model interpretation deals with issues facilitating the interpretation of results by modelers and decision makers. Of particular importance is the analysis of the sensitivity of model results to parameter variations, as well as sensitivity of the model to structural changes in the model, thus enabling closer inspection of model structure. Supporting the modeling life cycle in particular, and the need for providing more expressive power to models in general is research on more explicit and expressive model representations such as the Structured Modeling (Geoffrion, 1987; Muhanna & Pick, 1994).

Model management as a term was coined back in the mid-1970s, noting the importance of managing models and the modeling life-cycle in DSS (Sprague & Watson, 1975). Model management advances have often followed the advances in database management, due to the analogy between managing models and data (Dolk, 1986). In this viewpoint, models are treated as "black boxes" with a set of named inputs and outputs, and the goal of model management is that of insulating the user from intricate details of storing and processing models (Dolk & Konsynski, 1984). In other words, models are considered as data that need to be closely managed for integrity, consistency, security, and currency (Dolk, 1986) using what are known as model management systems.

Model management systems (MMS) are computer based systems that aid in the creation, storage, retrieval, manipulation, and utilization of models for decision makers. The goal of MMS is to facilitate problem solving by relieving the decision maker of coding algorithms and specifying models in procedural syntax (Liang, 1988). In essence, MMS provides a way to access and manage various modeling resources and the modeling life cycle. These resources include specific solvers (special algorithms or processes for solving specific problems), modeling platforms (software for developing and analyzing agent-based models), modeling languages such as general algebraic modeling systems (GAMS) and a mathematical programming language (AMPL), test files representing model schemas as used in GAMS models and MATLAB (numerical computing environment and programming language) models, and executable models.

Figure 1. Modeling lifecycle (adapted from Krishnan & Chari, 2000)