An Architecture for Organizational Decision Support Systems (ODSS) That Utilizes a Model Coordination Subsystem

GYEUNG-MIN J. KIM, Morgan State University, USA
JAMES R. BURNS, Texas Tech University, USA
ROHAE MYUNG, Morgan State University, USA

Organizational Decision Support Systems (ODSS) aim to support organizational decision-making that cuts across functional boundaries in the organization. As opposed to the conventional top-down approach to ODSS development, this research views integration of end-user-developed, organization-wide functional decision support systems as a viable approach for the development of an ODSS. First, this research examines coordination issues of those individual DSSs when they participate in organizational decision making as constituents of an ODSS. The coordination issues are discussed from the perspective of cycle time and quality of the organizational decision outcomes that the ODSS intends to promote. Then, a software architecture for ODSS that is capable of delivering coordination mechanisms for individually-developed DSS is presented. This work will be of interest to end-users and researchers involved with the fabrication and deployment of ODSS. In addition to the architecture, they will find important insights on how best to construct such software entities in the paper.

The development of PC-based applications by end-users is one of the most significant developments in organizational computing. As Gerrity and Rockart (1986) emphasized, discovering informal end-user applications with critical value to the organization and converting them into formal organizational systems could enhance organizational performance. This study identifies the possibility of integrating end-user developed decision support systems into a framework of organizational decision support (ODSS) in order to improve performance of organizational decision making.

The organizational decision process is a sequence of decision-making activities of the individual decision units. The decision units engage in modeling activities so as to solve their individual decision problems. The resultant models are used as a decision support system within the decision units. As an example, a collection of decision models developed by the sales unit becomes a Sales Planning DSS. Similarly, decision models developed for various aspects of production and corporate planning become Production and Corporate Planning DSS, respectively. Even though each DSS node is constructed and operated individually by the decision units, they, when tied together, collectively make up a Planning ODSS supporting the organizational planning process.

The key notions in the various ODSS frameworks are distributed problem solving by human and machine knowledge processors, communication among these problem solvers, and coordination of interrelated problem solving efforts in the interest of solving an overall decision problem (Holsapple and Winston, 1996). In this study, an ODSS refers to a collection of individual DSSs that communicate with each other to support collective organizational decision making. Each constituent DSS is dependent upon the whole but capable of supporting local decision-making. While the purpose of the individual DSS is to improve performance of individual decision-making, the goal of ODSS is to improve performance of collective organizational decision-making.

When multiple DSSs need to be integrated to form an ODSS, the following questions should be raised from the
holistic point of view. Does the performance of the individual DSS nodes ensure the performance of organizational decision making that is the collective outcome of the individual nodes? For example, does the quality of the individual DSS nodes ensure the quality of the organizational decisions that are the collective, aggregate outcomes of the individual nodes? Whenever interdependencies between constituent DSS nodes exist, the mechanisms to coordinate the individual DSS nodes play a vital role in making a reliable whole (Kim and Burns, 1998). In a similar way, optimum cycle-time of organizational decision making is ensured only when individual DSS nodes to support the constituent decisions are well coordinated and integrated as a whole (Kim, 1996).

In the example of a planning ODSS, the lack of coordination across DSS nodes could result in assumptions and scales of data used in the models across DSS nodes to be widely variant. An example is the difference in assumptions and scales of data used in the sales forecasting and production planning models. As a consequence, the production capacity mismatches with the sales capacity. When the mismatch is found, another iteration of the planning process is imperative. The reiterations required to achieve alignment result in longer cycle-times for organizational planning decisions. This problem becomes more expensive to remedy when it is found late in the process where the risk of dissatisfying customers is the greatest (because the production of the product cannot meet the customer’s deadline).

One approach to reduction of inconsistencies among models in Model Management Systems (MMS) is to facilitate sharing the same models. Such sharing is accomplished by establishing a centralized repository in which models are cataloged. Such a repository is sometimes referred to as a model base. While MMS standardize models intended for the same or similar purposes, it overlooks inconsistencies among decision models intended for different but related purposes such as models for sales forecasting and production planning models. The objective of this article is to present a software architecture for ODSS that provides various coordination mechanisms for integration of end-user developed decision support systems. The development approach for ODSS taken here is different from a top-down approach taken by others (Holsapple and Whinston, 1996; Carter et al., 1992). The top-down approach takes the view that ODSS development is not amenable to end-user development, thus ODSS should be developed by IS professionals. In this study we suggest a partnership approach in which the ODSS is developed through integration of individual DSS nodes developed by end-users with collaborative support coming from the IS professionals. The role of the IS department is implementation of coordination mechanisms among DSS nodes and other ODSS capabilities such as management of timing in data sharing among DSS nodes.

Finally, the architecture proposed here has important implications for the issues on knowledge management in organizations. Business organizations are moving toward knowledge-based ones where the center of gravity in employment is moving fast from manual and clerical workers to knowledge workers. To remain competitive, maybe even to survive, businesses will have to convert themselves into organizations of knowledgeable specialists. These knowledge workers develop their own decision support systems to direct and discipline their performance. Various types of functional decision support systems constantly create new information such as budgets, forecasts, recommended resource allocations that are then shared across the entire enterprise. Building a successful business intelligence environment requires a common knowledge management framework within which a variety of decision support applications are coordinated and integrated seemlessly. Such is the subject of and the motivation for this paper.

In order to fulfill the objective of this research, the following research questions are put forth:

1) What are the specific coordination problems among individual DSS nodes that must be resolved to ensure performance of collective organizational decision-making?

2) What are the appropriate mechanisms that solve the coordination problems identified in (1)?

3) What is a suggested architecture for the ODSS that is capable of delivering the coordination mechanisms identified in (2)?

The subsequent two sections investigate the first two questions. Then, after brief discussion of a generic architecture of ODSS, the third question is addressed. Finally, the conclusion of this study is presented.

**Types of Interdependencies and Associated Coordination Problems**

Coordination is defined as the act of managing interdependencies between activities to achieve a goal (Malone and Crowston, 1990). Interdependencies exist whenever the output(s) of one DSS model are the input(s) of one or more other DSS models. Interdependencies also exist when several DSS nodes are using models for essentially the same purpose; such models would have essentially the same inputs and outputs. Coordination problems occur when the interdependencies between decision models among individual DSS