Decision Process Cycle Time Reduction through Coordination of Modeling Activities

by

Gyeung-min J. Kim
Ewha Womens University

James R. Burns
Texas Tech University

PC-based modeling activity by end-users, used for decision support, is dependent upon information from other models and sources outside the functional unit. Consequently, there is a need for coordination of the various functional unit models that comprise the distributed decision support system. To the extent that the information sharing among these models is timely, current and correct, the resultant outputs of the models will be vastly improved. This article investigates coordination issues of end-user models with a focus on how such coordination affects cycle time of the organizational decision process. The article first discusses the various types of model interdependence and then presents a coordination strategy for accommodating each interdependence type. Significant cycle time reduction in organizational decision making processes can be achieved through use of one or more of the proposed coordination strategies.

The PC-based modeling activity by end-users is one of the most significant contemporary developments in organizational computing. It is known that productivity of individual decision makers is improved as PC-based modeling tools allow users to make more decisions within the constraints of cognitive, time and economic limits. The productivity improvement of individual decision makers has been a main focus of decision support systems research (Bostrom et al., 1990; Rivad et al., 1988; Sein et al., 1987) under the assumption that maximum performance of individual decision making ensures maximum performance of collective organizational decision making.

PC-based decision models are constructed and operated individually by functional units, yet these models are interdependent. For instance, the decision models in the sales and production functional units may be developed and used independently, yet they are interdependent in the sense that the assumptions and scales of data used in the sales forecasting and production planning models have to be the same. The lack of coordination in this interdependency could result in the production capacity being mismatched with the sales capacity. When the mismatch is found, another iteration of the planning process is imperative. The reiterations to achieve alignment result in longer cycle time of the organizational planning decision. This problem becomes more costly when production shortages result in unreliable promises and annoyed customers, and overages result in high inventory costs. Whenever interdependencies between constituencies exist, coordination of those constituencies plays a vital role for optimum performance of the whole (Davenport et al., 1990).

The primary objectives of this paper are to (1) understand various coordination problems in PC-based modeling activities that can lengthen the cycle time of organizational decisions, and (2) present various coordination mechanisms that can reduce this cycle time. A focus on cycle time not only drives down cost, but also leads to the
identification of defects, the elimination of which reduces the amount of rework and hence reduces cycle time. As the Internet economy drives a business revolution, the ability to make crisp decisions was rated first in importance in a study conducted by the Harbridge House in Boston (Turban et al., 1999). Additionally, this research helps both practitioners and researchers define a new role of the IS function related to end-users. This new role involves the coordination of end-user modeling activities with a view toward reduction of decision process cycle time.

Types of Interdependencies and Coordination Mechanisms

Coordination is defined as the act of managing interdependencies between activities to achieve a goal (Malone et al., 1990). In this section, interdependencies among decision models are identified and an appropriate coordination mechanism for each type of interdependency is discussed.

Common Model Interdependency

Interdependencies exist when several functional units need to use the same models for essentially the same purpose; such models would have essentially the same inputs and outputs. This type of interdependency is called common-model interdependency. In the context of a budget process, common-model interdependence is found among the operating budget models of the functional units. Every functional operating budget model has generic budget items like salaries, benefits, and supplies, and generic procedures to calculate those items. The lack of coordination in the common-model interdependency creates differences among the operating budget models. The causes of inconsistency are the following: assumption differences; differences involving data such as scale, type, name differences and missing or conflicting values; and procedure (business rule) differences. When the inconsistency is found later, another iteration of the planning process model template can be provided and shared by the functional units. Each functional unit extends the standardized model as necessary.

Prerequisite Interdependency

Interdependencies among decision models also exist whenever the output(s) of one model are the input(s) of one or more other models. This type of interdependency is called prerequisite interdependency. In the budget process, prerequisite interdependencies are found among the sales, production and the corporate budget models; the production budget model uses unit sales data from the sales budget model to determine unit-cost of goods sold that is used by the corporate budget model. Without electronic coordination, data delivery among decision models is through memo or report, followed by manual data retyping by the subsequent functional units. The lack of coordination leads to non-value-added activities such as memo writing and data entering activities that add extra time to an organization’s planning process. In addition, the hands-off between decision models causes longer wait time among process steps. Manually retyped data always has the potential to generate typing errors. The data errors that occur during retyping may introduce additional errors when the outputs of the model are passed on to the next phase of the process. When the errors are found later in the process, another iteration of the contaminated parts of the process is essential. The earlier the undetected error occurs, the more generations of errors it will produce, and thus the longer rework time it will end up causing, adding significant non-value-added time to the cycle time of the decision process. Coordination of the prerequisite interdependency involves (1) establishing standardized interfaces specifying communication requirements among related models, and (2) setting up a plan or schedule for sharing information among related models.
Concurrent Interdependency

Concurrent interdependency is defined when the data sharing among the prerequisite interdependent models occurs at the real time while the models are being executed. At the runtime, sensitivity analyses are conducted across distributed models, each of which represents the activities of a functional unit and collectively represent a series of interrelated flows of decisions among functional blocks in the firm. The following illustrates how companies are evaluating their planning strategies when concurrent interdependencies across models — the sales, production, and corporate budget models, for example — are not recognized and not coordinated. An additional model called a corporate planning model in which the interrelated flows of decisions across functional boundaries are modeled is used to investigate the consequences of alternative planning strategies on functional- and corporate-level performances. Development of a corporate planning model is a non-value-added activity that adds extra time to an organization’s planning process.

Coordination of the concurrent interdependence involves reuse and concurrent executions of existing models—the sales, production, and corporate budget models. Specifically, coordination of the concurrent interdependence involves (1) establishing standardized interfaces among related models, and (2) synchronizing model executions with the transmission of real-time information on alternative strategies to the subsequent (units). The synchronization coordination produces an organizational plan with a shorter cycle time not only by eliminating the time for developing the corporate planning model, but also by electronically transmitting outputs of the models as required to satisfy the concurrent dependency. In addition, synchronization coordination enables management across wider geographical boundaries to participate in the planning process through their respective models, thereby generating more rapid feedback on alternative strategies. Table 1 summarizes types of interdependencies and corresponding coordination methods.

An Internet Implementation of the Distributed DSS

Fortunately, the distributed decision support system being discussed has the ideal infrastructure to support it in place now — the corporate intranet or extranet. Each departments’ model is run on the departmental intranet node where it would be encapsulated with its local data. That node would have the capability to act as a client and initiate data requests of other departmental nodes that can act as servers. When a particular model requires the services of another model, it would initiate a request to the appropriate departmental node. That node’s model would then execute, provided it had no prerequisite dependencies and return the results back to the requesting node. If it had dependencies, it would initiate requests from still more nodes for resolution of those dependencies. Once resolved, it would then execute and forward its results back to the requesting node. In just milliseconds the entire distributed decision support system could be exercised and results presented in real-time to fulfill any information need.

One germane example of the need for the distributed decision support construct described here would be that of a customer reservation system. The customer wants to know when a certain quantity of a product can be delivered to their location. A distributed DSS set up to resolve this question might entail running models in the marketing, production, procurement, and
distribution departments. Indeed, the models of some suppliers might also have to be invoked to definitively arrive at an answer. The response comes back in less than a half-second of response time—8 days, say. At this point the customer can now make the decision whether or not to order this quantity of product. The distributed DSS provides all the essentials for an Internet-based customer delivery-date-determination and reservation system. This, in turn, is a way for companies to differentiate themselves from their competitors in what is now recognized as a most important competitive dimension—customer service. Differentiation is one of just three approaches to competitive advantage (Wetherbe, 1998; Porter, 1980).

Still another closely related example of distributed DSS would be that of integrated inventory management across components of the supply chain. Inventory management involves balancing customer service levels against the cost of carrying inventory (Retzlaff-Roberts, 1998). Decision support systems that support multi-location inventory systems involve determination of (1) geographic location of parts, (2) stock-up-to level of parts in each location, and (3) the resulting inventory and logistics costs. Determination of the “best” inventory policy is cumbersome and requires a holistic modeling approach that takes into account all related issues and parameters simultaneously.

Conclusions

Aside from the content of the information itself, the two most important characteristics of decision support information are its accuracy (quality) and its timeliness. Following principles espoused by Wetherbe (1995), this study has determined that significant cycle time reduction can result from scheduling, paralleling, integrating and standardizing. In this paper, strategies for improvement of both quality and cycle time have been discussed relative to the above-named principles. The exact extent of the improvement is reported in the simulation study of distributed budget processes (Kim, 1996).

According to the simulation study, the scheduling and synchronization coordinations reduce cycle time of the budget process by 72.35% and 6.5%, respectively, and together they reduce the cycle time of the budget process by nearly 100%. The results of the study confirm and validate what intuition would suggest about the scheduling and synchronization coordinations, namely that cycle time is shortened when scheduling and synchronization coordinations are applied. However, the results on standardization coordination seem to contradict the intuition that standardization coordination also reduces the cycle time of the process. This may be attributed to an assumption used in the simulation about organizational modeling activities. All of the

<table>
<thead>
<tr>
<th>Type of Interdependency</th>
<th>Coordination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Model</td>
<td>Standardization</td>
</tr>
<tr>
<td>Prerequisite</td>
<td>Scheduling</td>
</tr>
<tr>
<td>Concurrent</td>
<td>Synchronization</td>
</tr>
</tbody>
</table>

Table 1. Types of Interdependencies and Corresponding Coordination Methods
departments are assumed to wait upon the centralized IS staff to develop the standardized model template. This takes at least as long to develop as would non-standardized models developed in parallel by each department separately. As soon as the development of the standardized budget model template is completed, it is downloaded to all of the departments and all the common-model independent functional units start to extend the template simultaneously. Therefore, the time and effort saved in each functional unit, due to the standardization coordination method, could not be aggregated and reflected in the cycle time. Even so, it is readily apparent that considerably less effort would be required to deploy a standardized model template, rather than have each department develop its own model, which may then be inconsistent with similar models developed by other departments.

The decision to standardize the decision models has to consider the following factors: (1) development time for the standardized model, (2) stability of the model logic (i.e. how long the model logic remains unchanged), and (3) the number of users of the standardized models. The organization’s model administrator must evaluate the first factor against the second and third factors to decide which models should be standardized and kept in the organization’s model bank. There are, nevertheless, the benefits of non-duplication of effort among the various functional units endeavoring to produce similar models (which they must do without standardization coordination), and the benefit of avoidance of errors due to dissimilarities in those models. The simulation study (Kim, 1996) also indicates that the standardization coordination improves the quality of budget process by 62% through eliminating inconsistent outputs emanating from the process. Therefore, an additional factor that must be considered for standardization of decision models is the quality of the decision outcomes that the models support.

In effect, this paper has suggested various coordination strategies for re-design (reengineering) distributed decision support system processes, taken in relation to decision-process cycle time reduction. Particularly scheduling coordination, in which information is electronically transferred between origin/destination pairs, has a significant impact when the process is re-designed (Kim, 1996). Scheduling coordination will minimize the number of handoffs to humans that are required. Because of the non-value-added delays associated with handoffs to humans (queue times, setup times, wait times) and because human processors are orders-of-magnitude slower than computers, it follows that such handoffs should be minimized. Whenever humans were developing models, reworking models, typing reports, transmitting reports, and re-keying data, substantial amounts of time were consumed while the possibility for introducing errors increased drastically. Consequently, a basic principle for distributed decision support process design becomes manifest — minimize the number of handoffs to humans that are required if cycle time is a vital concern.

A focus on cycle time not only drives down cost, but also leads to the identification of defects, which must be eliminated for fast cycle time and quality decisions. In the planning example, as the cycle time increases, the accuracy of the organizational plan declines. In other words, as time lengthens, sales forecasts that guide the planning no longer reflect the current market condition. As Stalk emphasized in his time-based competitive strategy, longer cycle time of the decision process could have the following unforeseen impact: “with more forecasting errors, inventory balloons, and the need for the safety stocks at all levels increases, errors in forecasting mean more unscheduled jobs that have to be expedited, thereby crowding out scheduled jobs” (Stalk, 1988).

The coordination mechanisms proposed in this paper utilize both bottom-up and top-down strategies. The bottom-up approach takes the view that users in functional areas have the best knowledge about the functional requirements of their information systems. Therefore, it encourages
users to continuously develop their decision models, yet controls users' modeling activities in a way that they can be integrated with other models without conflicts. The bottom-up approach also ensures that the decision models are adaptive to the constantly changing business requirements of their respective domain.

The coordination methods proposed in this paper also take a top-down approach in that the coordination decisions across decision models are originated from the IS department that oversees interdependencies of functional information systems and can be implemented by building a hierarchical coordination system. Therefore, both approaches represent a synthesis of the functional, business management perspectives of end-users with the organization-wide, information resources management perspective of the IS department. Thus, a partnership between end-users and the IS department develops, which is considered to be more effective than centralized or reactive end-user management strategies (Gerrity, 1986).

References


Gyeung-min Joan Kim, Ph.D., is currently a professor at Ewha Womans University in Seoul, Korea. Her research interests include intelligent ERP systems, system integration methodology for electronic business, distributed decision support systems, and participatory design. She endeavors to align and adapt her research focus to the specific needs of the information technology and process management communities.

James R. Burns, Ph.D., is a Professor of Information Systems and Management Science in the College of Business Administration at Texas Tech University and formerly a research engineer with The Boeing Company in Seattle. Much of Dr. Burns’ published research is concerned with methodology for modeling continuous, lumped dynamic processes. He is interested in consolidation and reconciliation of various methodologies for enterprise integration. This interest is consonant with his long-standing work in the areas of process modeling and simulation. One specific focus is the integration of various modeling constructs into systems for decision support.
Decision Process Cycle Time Reduction through Coordination of Modeling Activities