International Journal of Policy Analysis and Information Systems, Vol. 6, No. 4, 1982

Optimal Design of a DSS System[1]

R. Preston McAfee[2] and Andrew B. Whinston[3]

Submitted September 1, 1982; accepted September 8, 1982

The paper develops a dynamic model of an interconnected multiprocessor system. Problems to be solved enter the system and are successively processed by different machines until a solution to the problem is achieved. A probabalistic characterization of problem transaction between processors is assumed. A steady state solution is determined which is then used in an optimization model to determine capacities for processors. An application of this model to office automation is presented.

KEY WORDS: Decision support systems; office automation; dynamic model.

1. INTRODUCTION

Most of the work in the area of Decision Support Systems (DSS) has concentrated on the development of micro models, i.e., models of a specific DSS. For example in Stohr,[5] the framework for a DSS oriented to an operation research environment was described. Generally, the micro models tends to focus on the problem of one processor helping an individual make a decision. Once the computerized system comes up with a solution, the results are passed to the decision maker, who then proceeds to take more action based on the information provided. Thus the typical DSS framework does not formally include the individual decision maker as part of the model and is thus generally limited to one processor.

The purpose of this paper is to focus on macro DSS issues. We will describe a DSS consisting of interconnected problem processors that will solve, over time, a series of problems. Specific problems are processed by a succession of processors until a solution is achieved. A dynamic model describing the behavior of this problem-solving system as a function of the

[1]Research was supported in part by NSF Grant Number IST-810-8519.
[2]University of Western Ontario, London, Canada N6A 5C2.
[3]Purdue University, West Lafayette, Indiana 47907.

0195-9301/82/1200-0425$03.00/0

capacity of each processor is presented. The steady state behavior of the system also depends on the capacity of the processors. In this we show how to determine the optimal capacities to minimize the net cost of operating the DSS.

The setting of an office provides an example of a typical integrated decision-making environment. Documents requiring various types of decisions are processed at workstations and passed on to another workstation for further processing until final action is taken and the document exits from the office. In this way, a network of workstations is produced. This network may be studied for optimal output characteristics. Some research has focused on bottleneck detection.[4] This paper proceeds further by optimizing capacity at each workstation, according to economic considerations and production possibilities. In this way, the bottleneck detection is extended to overall optimization. For an overview of office system topics, see Ref. 2.

2. GENERAL FORMULATION OF THE PROBLEM

Most research on distributed DSS has focused on either engineering and synchronization problems or the practical development of office machinery.[1] This has left a theoretical void in the optimization of such systems, filled by a few papers on bottleneck detection. One example is considered in Ref. 4. Part of the problem has been the complexity of modelling these systems in detail, and most research has focused on highly detailed studies of human/processor interaction. When the complexity of the synchronization is introduced, this level of detail makes a complete solution intractable.

One alternative is to explore a less detailed description of a multiprocessor environment. The particular example we employ uses the terminology of the office, however, it should be noted that the theory developed herein does not concern the office per se, but is applicable to a wide range of multiprocessor environments.

The multiprocessor network is a set of nodes, called workstations, or just stations, linked by paths of communication. For example, in Fig. 1, the Purchasing System of a firm links the firm departments via the documents that flow between departments. Workstations might also include human decision-making or forecasting models. The term documents is employed as a generic term and includes physical paper forms, electronic mail, telephone calls and even real goods.

The macroscopic view of the office permits averaging of the processing times and costs at each station. Thus the level of detail does not include individual document processing but only average processing rates, costs,

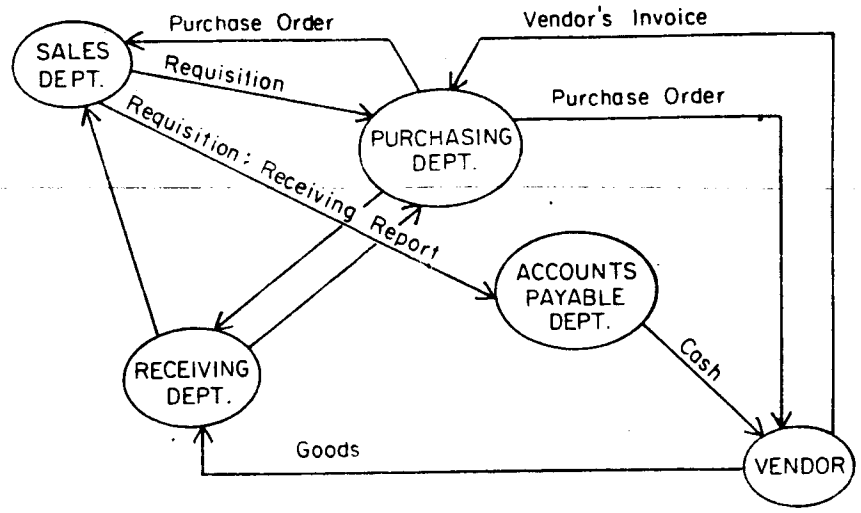

Fig. 1. Purchasing department linkage.

and times. In an office whose stations are largely homogeneous, this assumption is innocuous, as enough similar documents flow through the system so that the effects of any individual document are unimportant.

The office provides an excellent example of the macroscopic multiprocessor application, because duties at each workstation are relatively homogeneous. Moreover, the office provides an application of considerable economic importance, because office systems permiate business applications of machines. Thus the office provides an excellent setting for the model's application.

3. CONSTRUCTION OF THE MODEL

The model of the DSS system consists of k homogeneous document processing units, called workstations. These stations are denoted S_i, $i = 1, \ldots, k$. Each workstation S_i has a capacity c_i, which affects the rate at which documents are processed. At a given station, c_i may be the number of secretaries, clerks, bytes of RAM space, or floppy disk memory. At any given time, let a_i be the amount of available capacity, so that utilized capacity is $c_i - a_i \geq 0$. Each station will have documents being processed and others awaiting processing in a queue; the number of these documents at S_i is P_i and Q_i, respectively.

When a document has been processed, it is either sent to another workstation, or sent out of the system. The latter case includes destroyed documents. Define d_{ij} so that $d_{ij}P_i$ is the number of documents which, on

average, flow from S_i to S_j in one unit of time, given that P_i documents are in processing at S_i. Similarly, define e_i so that $e_i P_i$ is the number of documents which disappear from the system out of station S_i. If documents are copied at S_i, this rate may be subtracted from e_i. Thus e_i need not be positive. The rates documents move from S_i are assumed to be linear in P_i because we expect processing activity to take place at a constant rate. Thus, if one doubles the number of documents being processed, we expect to double the rate they flow out.

Let r_i be the rate at which documents flow to S_i from external sources, other than other workstations. This can include documents spontaneously generated at S_i, as long as these are not created from other documents at S_i (e.g., efficiency reports). Documents created from other documents at S_i can be considered as copies and are accounted for by e_i. Let f_i be the amount of capacity per document which is necessary, on average, to process a document at station S_i. Assume capacity at S_i has a cost w_i, and that the size of the queue, Q_i, has an imputed cost v_i. For future reference, these definitions are summarized in Table I.

$3k$ differential equations determine the values of Q_i, P_i and a_i, $i = 1, \ldots, k$. To specify them, we must consider how quickly documents leave the queue to be processed. This clearly depends only on Q_i and a_i. We shall assume the rate documents leave the queue is given by $b_i Q_i a_i$.[4]

If we let $\dot{x}$ represent dx/dt for any variable x and time t, we then have:

$$\dot{Q}_i = r_i - b_i Q_i a_i + \sum_{j=1}^{k} d_{ji} P_j \tag{1}$$

$$\dot{P}_i = -e_i P_i - \left(\sum_{j=1}^{k} d_{ij} \right) P_i + b_i Q_i a_i \tag{2}$$

$$\dot{a}_i = -f_i \dot{P}_i \tag{3}$$

Equations (1-3) follow from the definitions of the variables used.

4. OPTIMAL DOCUMENT PROCESSING

We assumed that capacity c_i has per unit cost w_i, and that the queues carry an imputed cost v_i. Assuming these are the only costs of the system,

[4] It is reasonable to assume that if excess capacity is doubled, the rate it is utilized is doubled, forcing the rate the queue is depleted to be linear in a_i. It is less obvious how Q_i should enter this formulation. The expression chosen is a first-order approximation of more general ones. In addition, it may be justified by assuming that the rate at which processing agents at a workstation come into contact with a document in the queue depends on the size of the queue. This amounts to saying that the more documents available for processing, the quicker unutilized capacity will be connected to the document.

Table I. Definition of Symbols Used

Symbol	Interpretation
S_i	workstation i
c_i	capacity at S_i
a_i	unused capacity at S_i
P_i	number of documents in processing at S_i
Q_i	number of documents awaiting processing at S_i
d_{ij}	probability a document leaves S_i for S_j per unit of time
e_i	probability a document is destroyed per unit of time
r_i	rate at which documents flow to S_i from outside the system
w_i	cost of capacity at S_i
v_i	imputed cost of the queue at S_i
f_i	amount of capacity necessary to process one document at S_i
b_i	rate at which free capacity is taken up at S_i

we may characterize the information flow problem by:

$$\text{Minimize Cost} = \sum_{i=1}^{k} (c_i w_i) + \sum_{i=1}^{k} v_i \lim_{T \to \infty} \frac{1}{T} \int_0^T Q_i \, dt \tag{4}$$

$$\text{subject to } c_1, \ldots, c_k \geq 0,$$

where

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T Q_i \, dt$$

gives the average size of Q_i. By the turnpike theorem,[3] we may assume that at some point $\dot{P}_i = \dot{Q}_i = 0$, and hence replace this limit with a value Q_i^*. The existence of Q_i^* depends on the possibility of stability of the system, which is guaranteed by Eqs. (1) and (2). If the c_i's are chosen too small, then the size of Q_i will diverge, but this would not minimize Eq. (4). It is important for the application of this theory to find Q_i^* and analogously, P_i^*. By Eqs. (1), (2) and the turnpike theorem, we have:

$$b_i(Q_i^* a_i) = r_i + \sum_{j=1}^{k} d_{ji} P_j^* \tag{5}$$

and

$$b_i(Q_i^* a_i) = e_i P_i^* + \sum_{j=1}^{k} d_{ij} P_i^* \tag{6}$$

Therefore, by Eqs. (5) and (6):

$$P_i^* \left(e_i + \sum_{j=1}^{k} (d_{ij}) \right) = r_i + \sum_{j=1}^{k} d_{ji} P_j^*$$

Letting

$$P^* = \begin{pmatrix} P_1^* \\ \vdots \\ P_k^* \end{pmatrix}, \qquad d_i = (d_{1i}, \ldots, d_{ki}),$$

$$\bar{r} = \begin{pmatrix} r_1/e_1 + \sum_{j=1}^{k} d_{1j} \\ \vdots \\ r_k/e_k + \sum_{j=1}^{k} d_{kj} \end{pmatrix}, \qquad \bar{d}_i = \frac{1}{e_i + \sum_{j=1}^{k} d_{ij}} d_i,$$

we have:

$$P_i^* = \frac{r_i + d_i P^*}{e_i + \sum_{j=1}^{k} d_{ij}}$$

Therefore if

$$\bar{D} = \begin{pmatrix} \bar{d}_1 \\ \vdots \\ \bar{d}_k \end{pmatrix}$$

$$P^* = \bar{r} + \begin{pmatrix} \bar{d}_1 \cdot P^* \\ \vdots \\ \bar{d}_k \cdot P^* \end{pmatrix} = \bar{r} + \bar{D}P^* \tag{7}$$

Thus $(I - \bar{D})P^* = \bar{r}$ and if $(I - \bar{D})$ is invertible, and $\delta_{ij} = \begin{matrix} 1 & i=j \\ 0 & i \neq j \end{matrix}$

$$P^* = (I - \bar{D})^{-1}\bar{r} = \left(\left(\delta_{ij} - \frac{d_{ij}}{e_i + \sum_{j=1}^{k} d_{ij}}\right)\right)^{-1} \begin{pmatrix} \dfrac{r_1^1}{e_1 + \sum_{j=1}^{k} d_{1j}} \\ \vdots \\ \dfrac{r_k}{e_k + \sum_{j=1}^{k} d_{kj}} \end{pmatrix} \tag{8}$$

Thus, P^* has been calculated in terms of d_{ij}, r_i, e_i, constants of the system. It is realistic to assume $d_{ii} = 0$, that is, station i does not transfer documents to itself.

We know that $(c_i - a_i) = f_i P_i$, because f_i is the amount of capacity per document (this follows from Eq. (3)). As a result:

$$c_i - a_i^* = f_i P^* \tag{9}$$

and by Eq. (5):

$$Q_i^* = \frac{r_i + \sum_{j=1}^{k} d_{ji} P_j^*}{b_i (c_i - f_i P_i^*)} \tag{10}$$

Thus Eq. (4) may be restated as:

$$\underset{c_i}{\text{Min}} \sum_{i=1}^{k} c_i w_i + \sum_{i=1}^{k} v_i Q_i^*$$

First order conditions yield:

$$w_i + v_i \frac{\partial Q_i^*}{\partial c_i} = 0$$

Therefore, since

$$\frac{\partial P^*}{\partial c_i} = -\frac{r_i + \sum_{j=1}^{k} d_{ji} P_j^*}{b_i} (c_i - f_i P_i^*)^{-2}$$

Therefore

$$c_i = \sqrt{\frac{v_i (r_i + \sum_{j=1}^{k} d_{ji} P_j^*)}{b_i w_i}} + f_i P_i^* \tag{11}$$

Equation (11) provides the optimum capacities c_i at each station. It is interesting to observe from Eq. (8) that the number of documents in processing is, on average, invariant to the processing capacity.

Equation (11) has a sensible interpretation, i.e., $c_i - f_i P_i^* = a_i$, the excess capacity. Reformulating Eq. (11), using Eq. (10), we have:

$$v_i Q_i = a_i w_i \tag{12}$$

This suggests that the cost of the queue equals the cost of available capacity noting that the differential relationships are linear.

5. ESTIMATION PROCEDURE

We have seen that under the hypothesis of nonsingularity of $(I - \bar{D})$, the model produces well-defined coefficients for P^*, Q^* and a^* as a function of d_{ij}, e_i, r_i, e_i, f_i and b_i. We expect that, examination of actual operating systems will provide estimates of P_i, Q_i, c_i, $c_i - a_i$ (utilization of capacity), r_i, e_i, and f_i (processing time spent on one document). In addition, w_i is directly observable as it is only a price. This leaves d_{ij}, b_i and v_i. Because d_{ij} is the rate (probability) that documents leave S_i for S_j, $(1/d_{ij})$ is the expected length of time a document headed for S_j remains in state P_i, and is hence observable. This unfortunately means that the inverse of d_{ij} is estimated empirically (by averaging). This is unfortunate, as $E(1/x) > [1/E(x)]$ for nondegenerate distributions of x. Assumptions on distribution of processing times on documents could allow for a consistent estimation of d_{ij}, but we will merely assume the estimate of d_{ij} by observing the average $1/d_{ij}$ is reasonable. Equation (10) can be used to estimate b_i.

The variable v_i is the imputed cost of Q_i and is not directly observable. Consequently, this model could be used to calculate the values v_i and thus provide evidence on the value that the office manager places on the different queue sizes. It is not to be expected that the v_i's will be equal, as different queues have differing importance to the firm. By Eq. (11), we note that the optimal c_i does not depend on v_j for $j \neq i$. Thus, if some relationship on some subset of the v_i's is plausible, for example, it may be expected that $v_i = v_j$ for some i and j, then the efficiency of the office or applicability of the model may be tested. This occurs because the model will predict the relationship between c_i and c_j as a function of the relationship of v_i and v_j, and consequently this relationship is testable. The model also predicts that c_i is linear in $(w_i)^{-1/2}$. Because w_i (the price of c_i) and c_i are observable, another test of the model's applicability would be a regression on c_i and $w_i^{-1/2}$. Unfortunately, the lack of such a relationship does not necessarily discredit the model, but might merely indicate information problems or inefficiency.

Finally, Eq. (12) provides a simple way of associating the value imputed to shortening the queue. The total cost of excess capacity at S_i equals the imputed cost of the queue. Thus, a simple rule permits the firm to evaluate if it is not spending the right amount at S_i. If the cost of the queue appears to be less than the cost of the excess capacity available in the system, then the firm is spending too much, and conversely. This is such a simple rule of thumb that it may be useful in actual situations.

6. CONCLUSION

This paper develops a macro-scaled model of an office information system, and solves for the optimal capacities of each workstation. The theory is applicable to a wide variety of decision support systems beyond the office model explicitly considered.

A similar approach is used by Ladd and Tsichritzis,[4] in the sense that average document flow is examined in a directed graph. They proceed to detect bottlenecks in the system. The problem with this is that the existence of a bottleneck in a DSS system does not indicate suboptimal planning. Indeed, if there are no queues at all, it is likely that too much capacity has been used. For this reason, we considered optimizing the capacity at each station, given a capacity cost and an imputed cost to queues.

A number of possible extensions of this paper are possible. First, we have not considered the impact of peak loads on the system. A more complete treatment would permit stochastic variation in the inputs to the system and in the processing times, and optimize capacity given this. The difficulty here is that the objective function of the decision maker is hard to specify except in an ad hoc fashion.

Another extension would involve more general formulations of the rate at which documents in the queue are transferred to processing. In addition, some empirical investigation of DSS systems might suggest different models of document processing that could then be fine tuned in a way consistent to our approach.

REFERENCES

1. A. Bailey, J. Gerlach, R. McAfee, and A. Whinston, "Office automation," *Handbook of Industrial Engineering* (Wiley and Sons, forthcoming).
2. A. Bailey, J. Gerlach, R. McAfee, and A. Whinston, "An "OIS model for internal control evaluation," *Transactions on Office Information Systems* (forthcoming).
3. M. Intriligator, *Mathematical Optimization and Economic Theory* (Prentice-Hall, Inc.: 1971), pp. 435.
4. R. Ladd and D. Tsichritzis, "An office form flow model" (unpublished manuscript).
5. Edward A. Stohr and Mohan R. Tanniru, "A database for operations research models," *International Journal of Policy Analysis and Information Systems* **4**(1) (1980).