Fong, Elizabeth N., et. al., 1985. Guide on Logical Database Design. National Bureau of Standards Special Publication 500-122. Washington, DC: U. S. Government Printing Office.

Guevara, Jose Armando, 1983. A Framework for the Analysis of Geographic Information System Procedures: The Polygon Overlay Problem, Computational Complexity and Polyline Intersection. Unpublished Ph. D. dissertation, Department of Geography, State University of New York at Buffalo.

Hawthorn, Paula, 1985. "Variations on a Benchmark," Database Engineering, Vol. 8, No. 1, pp. 19-28.

Heidelberger, Philip and Stephen S. Lavenberg, 1984. "Computer Performance Evaluation Methodology," IEEE Transactions on Computers, Vol. C-33, No. 12, pp. 1195-1220.

Letmanyi, Helen, 1984. Assessment of Techniques for Evaluating Computer Systems for Federal Agency Procurement. NBS Special Publication 500-113. Washington, DC: U.S. Government Printing Office.

Sheppard, Charles L., 1985. Guide for Selecting Microcomputer Data Management Software. NBS Special Publication 500-131. Washington, DC: U. S. Government Printing Office.

Stonebreaker, Michael, 1985. "Tips on Benchmarking Data Base Systems," Database Engineering, Vol. 8, No. 1, pp. 10-18.

_____ (ed.), 1986. The INGRES Papers: Anatomy of a Relational Database System. Reading, Mass.: Addison-Wesley Publishing Co.

Tomlinson, Roger F. and A. Raymond Boyle, 1981. "The State of Development of Systems for Handling Natural Resources Inventory Data," Cartographica, Vol. 18, No. 4, pp. 65-95.

Yao, S. Bing and Alan R. Hevner, 1984. A Guide to Performance Evaluation of Database Systems. NBS Special Publication 500-118. Washington, DC: U. S. Government Printing Office.

_____, 1985. Benchmark Analysis of Database Architectures: A Case Study. NBS Special Publication 500-132. Washington, DC: U. S. Government Printing Office.

PERFORMANCE EVALUATION AND WORKLOAD ESTIMATION FOR GEOGRAPHIC INFORMATION SYSTEMS

Dr. Michael F. Goodchild
Department of Geography
University of Western Ontario
London, Ontario, Canada
N6A 5C2

Brian R. Rizzo
Canada Land Data System
Lands Directorate
Environment Canada

ABSTRACT

Agencies acquiring GIS hardware and software are faced with uncertainty at two levels: over the degree to which the proposed system will perform the functions required, and over the degree to which it is capable of doing so within proposed production schedules. As the field matures the second concern is becoming more significant. A formal model of the GIS acquisition process is proposed, based on a conceptual level of GIS subtask definition. The appropriateness of the approach is illustrated using performance data from the Canada Land Data System. It is possible to construct reasonably accurate models of system resource utilization using simple predictors and least squares techniques, and a combination of inductive and deductive reasoning. The model has been implemented in an interactive package for MS-DOS systems.

INTRODUCTION

The development of geographic information systems has now reached the point where substantial numbers of turnkey production systems are being acquired from vendors and installed in public and private sector agencies. In many cases these agencies will have made detailed plans for the use of the system prior to its selection, including evaluation of workloads, and will have required potential vendors to respond directly to these plans. The vendor in turn will have provided information on the extent to which the proposed system is capable of performing the prescribed work, both in terms of specific functions, and in overall utilization of system resources.

In many ways this ideal, objective and precise model of the system acquisition process is rarely achieved in practice. The agency must first identify the precise products which it expects to obtain from the system over the planning period, and the

system products. For example, it is relatively easy for forest managers to define an updated forest inventory map as the result of overlaying recent fire polygons on existing forest inventory polygons, but any lower level of subtask definition would presume substantial familiarity with one or more GIS's.

There is a long history of debate in the performance evaluation field over the extent to which one should regard the system as a black box, observing the response of the system to given inputs in a purely empirical context, or whether the approach should be to some degree determined by knowledge of the algorithms being used. For example, we might expect the major factor determining execution time in a raster polygon overlay algorithm to be the raster cell size, whereas a vector algorithm would be more likely to depend on polygon counts. Lehman (1977) makes this point, and notes that the need for empirical, black box performance evaluation is in fact somewhat paradoxical since the system under study is in principle perfectly understood. An interesting commentary on the field by Wegner (1972, p. 374) urges "a proper balance between quantitative statistical techniques and qualitative techniques of structural analysis", although, somewhat surprisingly:

> "Computer science is different in character from empirical disciplines such as agriculture or physics. Agriculture and physics are concerned with the study of natural phenomena, while computer science is concerned with the study of man-made phenomena. A computer system generally has a far larger number of independently variable components than the systems studied in agriculture or physics".

The debate would seem to be more complex in the GIS field where there is no control over the choice of algorithm used to perform a given subtask, and where some of the operations being modelled are manual or contain substantial manual components. For example, it is essential to have a satisfactory model of digitizer throughput, including operator time spent correcting errors, if one is to make adequate projections of the number of digitizer shifts necessary to complete a given workload. In fact this has been one of the more uncertain elements in many GIS acquisitions.

There is of course no chance than predictions of system utilization made from the results of performance evaluation will be perfectly accurate. Many of the factors influencing throughput cannot be predicted in advance, and others can be predicted only with considerable uncertainty. Obvious candidates in the first category are various types of hardware and software failure. The task is best seen as a compromise between an excessively elaborate model on the one hand, which would require too much data collection and rigid adherence to planned production schedules, and would be too sensitive to uncertainties, and on the other hand too little effort at assessing the degree to which the planned workload lies within the capacity of the proposed

system. We assume that the alternative of no prior workload evaluation is unacceptable.

The empirical or statistical approach to performance evaluation has been discussed in a number of articles (see for example Gomaa, 1976; Grenander and Tsao, 1972; Yeh, 1972; Bard and Suryanarayana, 1972; Racite, 1972), and the associated problems of experimental design have been discussed by Nelder (1979). The conventional technique is ordinary least squares regression, although Grenander and Tsao (1972) comment that its use cannot be too rigid since it is usually impossible to meet the inferential assumptions of the technique. Racite (1972) discusses the use of nonlinear regression.

FORMAL MODEL

We now present a formal model and notation for the acquisition and benchmarking process, following the conceptual outline given above.

The agency has defined a set of products $R_1, R_2 \ldots R_i \ldots$, each one in the form of a map or tabular printout, or some combination of the two, and each one requiring the execution of a sequence of GIS operations or subtasks. The number of each product type required in each year j of the planned period is denoted by Y_{ij}. The subtasks are defined by an ordered set which may include several executions of the same type of subtask, for example several polygon overlays. The subtask sequence for product i is denoted by:

$$S_i = \{S_{i1}, S_{i2} \ldots S_{it} \ldots\} \qquad (1)$$

with each subtask drawn from a library L, $S_{it} \in L$ for all i,t.

Each subtask a in the library is associated with a number of measures of utilization, drawn from a standard set M. Each measure m_{ak} represents some demand on the system, such as cpu time, operator time, plotter time or disk storage requirement, with appropriate units of measurement. The value of each measure for a given task can be predicted from one or more predictors p_{akn}, drawn from a standard set P. The predictors for each measure are quantities such as number of polygons which can be estimated in advance for each of the required products and used to estimate total resource utilization. Note that the set of predictors for a given measure may vary from subtask to subtask, but not from product to product. The predictive equations for each measure are functions:

$$m_{ak} = f(p_{ak1}, p_{ak2} \ldots p_{akn} \ldots) \qquad (2)$$

calibrated by least squares regression or other means. The precise choice of function will be determined by a combination of empirical investigation and analysis of the subtask structure.

To estimate system resource utilization, we examine the required subtasks for each product. The predictors for each measure are determined from the planned production schedule and used to evaluate the appropriate form of the predictive equation (2). The measures are then summed for the product as a whole:

$$W_{mi} = \sum_t W_{mit} \quad \text{for all } m_{ak},\ a=s_{it} \qquad (3)$$

and across products, weighted by the number required in each year:

$$V_{mj} = \sum_i W_{mi} Y_{ij} \quad \text{for all } m \qquad (4)$$

to give total resource requirements which can be compared to known capacities.

EMPIRICAL ANALYSIS

The Canada Land Data System (Canada Geographic Information System) (CGIS) was designed in the early 1960's as a system for input and analysis of a national land capability survey consisting of multiple layers of polygon data. Its most significant features are the use of a scanner for data input, conversion to vector organization for storage, and a raster algorithm for polygon overlay. Other features of the system will be noted during the discussion which follows. The data to be analyzed were collected during regular production as part of the everyday CGIS internal auditing process.

The data sets were all processed as part of a larger study of land use change in Canadian metropolitan cities. Four coverages were processed for each of 6 cities, Windsor, London, Kitchener, Hamilton, Regina and Montreal. All input was obtained from complete 1:250,000 map sheets, the number of sheets varying from 2 in the case of London to 9 in the case of Montreal. One sheet was shared between Hamilton and Kitchener so its input costs were incurred only once. In total 104 sheets were input, for each of 26 map sheets and 4 coverages.

Three major subtasks have been identified in the input process for the purposes of this study, and the resource utilization expressed in dollars. Before scanning, each input document must be copied by hand using a scribing tool, to control line width and to ensure against spurious input. The costs of scribing (SCRIBE) are largely those of labour, and can be assumed to depend on the length of polygon boundaries being scribed, and also to some extent on the irregularity of the lines and on the density of features. Following scanning the raster data is vectorized and merged with polygon attributes in processes referred to as steps 0 to 4, for which cost (denoted by Z4) is

primarily determined by computer use. CGIS processes its data through a service bureau, so the costs given are those billed by the bureau, as distorted by the peculiarities of the billing algorithm and such factors as overnight discounts. The third cost is that of manual error correction (MEC), which occurs during input processing, and consists of the labour required to identify and remove errors detected by software during vectorization and polygon building.

Only one predictor is available for the three subtasks, in the form of a count of the number of polygons on each sheet. Although many more sensitive predictors might be obtained from the data after input, such as counts of coordinate pairs or line lengths, it is relatively easy to estimate polygon counts for typical map sheets in advance.

The four coverages used in the study are as follows:

Code	Theme	Mean Polygon Count
040E,F	Study area outline	3.2
100E	Recreation capability	59.7
200E	Agriculture capability	238.5
760X	Land use change	1142.4

The theme of each sheet accounts for a large amount of the variance in input costs: 40.1% of SCRIBE, 45.3% of Z4 and 28.2% of MEC. But almost all of this is because of variation in polygon counts; although each coverage type has different conditions of polygon shape and line contortedness, disaggregating by coverage produces no significant improvement in our ability to predict costs once polygon counts have been allowed for.

The best fit was obtained by a double logarithmic or power law model of the form:

$$m = ap^b \qquad (5)$$

where a and b are constants, calibrated by regressing the log of each measure against the log of the predictor, in this case log(cost) against log(polygon count). The results are shown below:

Measure	Variance explained	b	Standard error of estimate
SCRIBE	84%	0.69	0.30
Z4	72%	0.31	0.19
MEC	68%	0.53	0.25

The manual operation of scribing has the most predictable costs in terms of variance explained. Assuming no variation in shape, on purely dimensional grounds we would expect the total length of polygon boundaries on a map sheet to be proportional to the square root of the number of polygons. However the regression shows that scribing costs rise with the 0.69 power, indicating that a higher density of polygons requires more effort per unit length of line than the added line length would suggest, due presumably to the added complexity of working with high densities.

We expect the vectorization steps to be relatively insensitive to the number of polygons, and indeed the calibrated power is the lowest at 0.31, indicating that a doubling of cost will allow for the processing of a sheet with approximately eight times as many polygons. MEC costs rise with the 0.53 power, suggesting either that the probability of error is dependent on length of line, or that the difficulty of correction is approximately twice as great for a sheet with four times as many polygons.

The standard errors of estimate are given above for each of the three sets of costs. Since the regression was performed on the logs of the cost values, a standard error of e must be interpreted as meaning that the error of prediction from the model is typically a factor of 10^e. In the case of SCRIBE, which has the largest standard error, the error factor is therefore 2.0, meaning that we will commonly observe actual scribing costs which are half or twice the predicted value. Although this is a substantial uncertainty, it is very much less than the range of map sheet scribing costs, which vary from a low of $2 to a high of over $2,000.

After completion of the input steps, including edgematching of adjacent sheets, the data were merged into six data bases, each with four coverages. The coverages were then overlaid using the CLDS polygon overlay algorithm, which employs raster techniques to superimpose vector data structures. Both cpu time and billed cost were available as measures for each overlay operation, the relation between them being proprietary to the computer service bureau, and compounded by CLDS job scheduling decisions. Linear regression of overlay cost on overlay time showed that only 74% of variance in cost is accounted for by variance in execution cpu time. Total input costs for each city's data were also available, but gave results which added little to those already obtained for the map sheet data: since the largest component of input cost is scribing, regression of total cost on polygon count gave results very similar to those shown above for SCRIBE.

The results of regressing log(overlay cost) and log(overlay time) on the logs of various polygon counts are shown below in terms of variance explained:

Count	Time	Cost
Total output	85%	31%
Total input	79%	27%
040E/F	80%	53%
100E	59%	44%
200E	81%	46%
760X	73%	21%

The increase in uncertainty introduced by the billing algorithm is clear in all cases. Not unexpectedly given the nature of the overlay algorithm the best predictor is total output polygon count, reflecting the cost of revectorizing the image after overlay and building new polygon attribute tables. The estimated power is 0.44, which compares well with the power of the Z4 vectorization above. The standard error of estimate is 0.14, or an error factor of approximately 1.4. Although output polygon count would not be available as a prior predictor of system workload, it is linearly related to total input count: for this data, each input polygon generates on average 2.54 output polygons, the input count explaining 85% of the variation in output count. The standard error of estimate if log(input count) is used to predict overlay time rather than log(output count) is 0.16 rather than 0.14.

From this analysis it appears to be possible, given stable software and hardware and sufficient data, to model the performance of a GIS at the level of the conceptual GIS subtask, and to obtain reasonably accurate predictions of resource utilization. As we noted above, there is no possibility of perfectly accurate modelling: on the other hand, any reduction in uncertainty is presumably better than pure guesswork in system planning. The same basic approach of curve fitting seems to be suitable equally for machine utilization as for purely manual or mixed manual and machine operations. The next section describes the operationalization of the model, including calibration steps, in an interactive package.

IMPLEMENTATION

The first author and Tomlinson Associates have implemented the formal model and calibration procedures discussed above in a package for MS-DOS systems identified as SPM. It is structured in 8 interdependent modules linked by a master menu, as follows:

Module	Function
1	Build, edit or retrieve the library of subtasks L.

2 Input ordinal performance scores for each subtask from the results of a qualitative benchmark test.

3 Input definitions for a set of required products $R_1, R_2 \ldots R_i \ldots$, including required processing steps.

4 Generate a statistical report based on the ability of the system to produce the required products, given the input performance scores.

5 Input values of suitable performance measures and predictors from the results of a quantitative benchmark test.

6 Construct and calibrate suitable models of each subtask from the data input in the previous step.

7 Input predictor values measuring intended system workload for each product.

8 Compute and generate a statistical report giving cumulative resource utilization estimates for the intended workload.

Module 6 allows the user to choose from a wide range of possible models, including additive and multiplicative combinations of predictors, and various transformations of variables. The values of constants can be obtained either by ordinary least squares or by direct user input.

A recent test of the approach used data obtained by Tomlinson Associates from a US National Forest GIS requirements study. The Forest staff had previously identified a total of 55 GIS products which they planned to use in their resource management activities in the first 6 years of GIS operation. The combined production task required a total of 65 coverages or data types to be input to the system, and a total of 51 different GIS functions or subtasks to perform the required manipulations. The number of subtask steps required for each product ranged from 5 to 24.

Because of the effort involved, benchmark performance models were constructed using SPM only for the 8 most resource-intensive subtasks, including polygon overlay, buffer zone generation and edgematching. Four measures were used: cpu time, personnel time, plotter time and disk storage bytes. The predictive models relied on a total of 11 different measures, including polygon, line and point counts as appropriate to each subtask. The final results were expressed in terms of total resource requirements for each product in each year of production, given the benchmarked hardware and software configuration.

DISCUSSION

Agencies acquiring GIS's have had to contend with considerable uncertainty, first over whether the system being acquired could indeed perform the necessary manipulations of spatial data, and secondly over whether the computing resources of the system were sufficient to meet required production schedules. GIS software has now reached a stage of development where much of the first form of anxiety has been removed: functions such as polygon overlay and buffer zone generation now perform with reasonable efficiency in most systems. However the models of system performance required to reduce uncertainty of the second type do not yet exist to any great extent.

The most critical step in modelling performance is the definition of subtask. The conceptual level of subtask definition used in this paper matches the level used for most GIS user interfaces, and is readily understood by agency staff not otherwise familiar with GIS operations and concepts. The empirical section of this paper has shown that it is possible to model performance at this level even though subtasks may include substantial manual components and may have to allow for unpredictable events such as hardware failure.

We noted earlier that any successful modelling effort must not simply approach a system as a black box, but use knowledge of the complexity of subtasks and GIS algorithms to anticipate appropriate predictor variables and their role in the form of predictive models. This point also applies to the design of benchmarks, since the same arguments can be used to make suitable choices of measures and predictors, and to design appropriate variations of the key parameters. The number of independent runs required to obtain a reliable calibration of a given model is also determined by the number of variables and constants appearing in the model: conversely, the choice of possible models is constrained by the number of independent benchmark tests made of each subtask.

In this paper we have assumed that the hardware and software configuration benchmarked is also the one proposed for production: no attempt has been made to develop models valid across configurations. To do so would add a new level of difficulty to the modelling which is outside the context of the present study. On the other hand the choice of the conceptual level for subtask definition allows the same general strategy to be followed whatever the configuration.

This last point restricts the applicability of this approach to the context defined in the introduction, that of a vendor or agency wishing to make a reliable estimate of resource utilization for a given workload and a given system. It is not useful for an agency wishing to make a comparison between alternative systems, except as a means of developing information which might later form the basis of the comparison.

ACKNOWLEDGEMENTS

We wish to acknowledge the assistance of the Canada Land Data Systems Division, Environment Canada in providing the data for this study.

REFERENCES

Bard, Y. and K.V. Suryanarayana, 1972. "Quantitative methods for evaluating computer system performance: a review and proposals." In W. Freiberger, editor, Statistical Computer Performance Evaluation. Academic Press, New York, pp. 329-346.

Beilner, H. and E. Gelende, editors, 1977. Measuring, Modelling and Evaluating Computer Systems. North Holland, New York.

Chandy, K.M. and M. Reiser, editors, 1977. Computer Performance. North Holland, New York.

Ferrari, D., 1978. Computer Systems Performance Evaluation. Prentice-Hall, Englewood Cliffs, New Jersey.

Gomaa, H., 1976. "A modelling approach to the evaluation of computer system performance." In E. Gelende, editor, Modelling and Performance Evaluation of Computer Systems. North Holland, New York, pp. 171-199.

Grenander, U. and R.F. Tsao, 1972. "Quantitative methods for evaluating computer system performance: a review and proposals." In W. Freiberger, editor, Statistical Computer Performance Evaluation. Academic Press, New York, pp. 3-24.

Hellerman, H. and T.F. Conroy, 1975. Computer System Performance. McGraw Hill, New York.

Jones, R., 1975. "A survey of benchmarking: the state of the art." In N. Benwell, editor, Benchmarking: Computer Evaluation and Measurement. Wiley, New York, pp. 15-23.

Lehman, M.M., 1977. "Performance evaluation, phenomenology, computer science and installation management." In K.M. Chandy and M. Reiser, editors, Computer Performance. North Holland, New York, pp. 1-16.

Nelder, J.A., 1979. "Experimental design and statistical evaluation." In L.D. Fosdick, editor, Performance Evaluation of Numerical Software. North Holland, New York, pp. 309-315.

Racite, M.P., 1972. "The use of pure and modified regression techniques for developing systems performance algorithms." In W. Freiberger, editor, Statistical Computer Performance Evaluation. Academic Press, New York, pp. 347-369.

Wegner, P., 1972. "Discussion of Section V." In W. Freiberger, editor, Statistical Computer Performance Evaluation. Academic Press, New York, pp. 372-374.

Yeh, A.C., 1972. "An application of statistical methodology in the study of computer system performance." In W. Freiberger, editor, Statistical Computer Performance Evaluation. Academic Press, New York, pp. 287-327.