BAYESIAN MODELS OF DESIGN BASED ON INTUITION

K.M.Chandy

IBM Thomas J Watson Research Center

(on leave from The University of Texas at Austin)

KEYWORDS: design, analysis, Bayesian decision theory, intuition,uncertainty, probability theory, distributed data base systems, dynamic programming, optimization

ABSTRACT: Most computer system designers use a great deal of intuition in the design process. Intuition is often used to handle uncertainty in design parameters. Since uncertainty seems to be intrinsic to most design problems it follows that designers will continue to rely on intuition or "sound engineering judgement". This paper attempts to use Bayesian Decision Theory to explore the possibility of setting up a structure and theory for making design decisions in the computer system design environment while explicitly taking the intuitive nature of many design decisions into account. We shall focus attention on a particular problem in distributed data base design in which the designer must use his intuition to estimate the load on the system which he is designing. Similar Bayesian approaches could be used in other design problems.

1. INTRODUCTION

1.1 Overall objectives This paper is concerned with the use of decision theory to model the design process in a computer system design. We shall focus attention on the design of certain simple distributed data base systems. Systems are generally designed under two environments:

(a)An existing system does not perform satisfactorily on a given workload and a new system is to be designed to handle the existing workload.

(b)No system exists and the projected workload is not known with certainty.

We shall be concerned with the second environment. Our objectives in this paper are to:

(a) cope with uncertainties in projected demand (workload) in a rational manner by using Bayesian Decision Theoretic tecniques, (Pratt, Raiffa and Schlaifer, 1965],

(b)develop models to evaluate various tradeoffs in certain simple distributed data bases in a simple yet quantitative manner.

Models are usually developed at several levels of detail. Analytic models are used to search a parameter space rapidly while detailed simulation models are used to obtain more precise estimates of system characteristics. The number of possible design configurations may be too large (possibly of the order of 100 million) to use simulation methods to evaluate each configuration. In this paper we restrict attention to gross analytic design models.

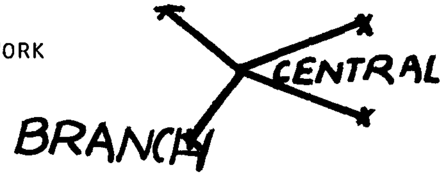
1.2 A qualitative description of the system

This paper is concerned with the design of distributed processing networkd with star topologies. Though such networks may be used in a variety of medium-scale organizations we shall concretize the problem by considering the design of a proposed network for a state Mental Health Mental Retardation (MHMR) agency

The organization consists of a central office and several branch offices. In the MHMR context, the central office is the state data processing headquarters and the branch offices are hospitals, mental health evaluation centers, schools and halfway houses. The geographical location of the central and branch offices are fixed. The organization's data processing network supports many independent data bases corresponding to different organizational functions. For instance in the MHMR context the data bases supported may include a pharmaceutical data base and a mentally retarded persons skills data base. The MHMR agency is required to support a pharmaceutical function and a mentally retarded person's skills analysis function and various other functions. The functions are independent in the sense that a given transaction pertains to one and only one function; therefore a given transaction utilizes one and only one data base. (In practice, the different functions are not strictly independent; however the large majority of the transactions are concerned with only one function).

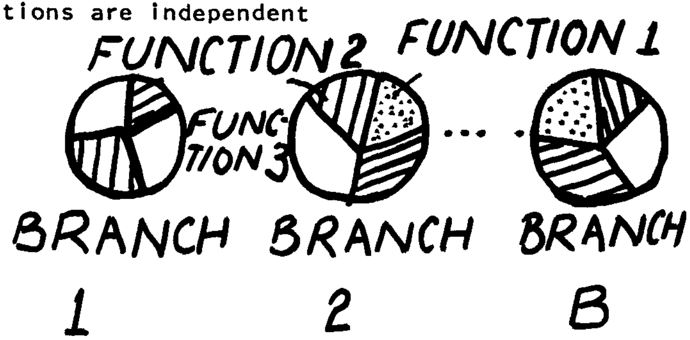
Each branch office carries out some or all of the functions carried out by the network. The data bases pertaining to a given branch office are independent of the data bases pertaining to all other branch offices. In the MHMR context, a branch office's data base is concerned with the patients being handled by that branch office. A branch office's data base will not normally contain information about patients being treated elsewhere. When a patient moves from one branch office to another his files are moved with him. A patient is likely to receive treatment at the same branch office for a considerable amount of time before he moves to another branch office; hence the proportion of transactions which are concerned with the movement of files from one branch office to another is small compared with the number of transactions directed towards one and only one branch office. Therefore, we shall assume that all the branch office data bases are independent.

Transactions generated at the central office may access any data base in the system since the central office has administrative authority over all branches.



A STAR NETWORK
figure 1

The data bases pertaining to the branch offices are therefore partitioned at two levels as shown in figure 2:
(a) data bases at different branches are independent and
(b)data bases corresponding to different functions are independent



INDEPENDENT DATA BASES
Figure 2

1.3 Given parameters

We are given:
(a) estimates of the rates at which transactions corresponding to different functions are generated at each of the offices
(b)distributions of message lengths and processing times required by different kinds of transactions
(c)a set of data processing systems from which we must select one for the central office
(d) for each branch office a set of data processing systems from which we must select one for that branch office
(e) for each link between a branch office and the central office we are given a set of communication lines from which we must select one for that link.

Estimates of message length and processing time distributions fpr specific kinds of transactions are relatively accurate since these estimates are usually obtained from measurements on existing (possibly prototype) systems. Estimates of transaction processing rates are based on projections of the future demand for data processing. These estimates tend to be soft in the sense that a designer is generally quite unsure about his estimates, since demand may be influenced by a variety of factors including the economy, government policy decisions and other factors over which the designer has no control. Furthermore, different members of the design team may have significantly different estimates. Uncertainty regarding load is probably the most common form of uncertainty. However many design decisions, such as whether to choose one memory technology or another, deal with more fundamental and more pervasive uncertainty.

1.3.1 Estimating demand

The design/marketing team benefits from working with several possible scenarios, with the assumption that one of the scenarios will develop. The "scenario space" is usually continuous; however for computational tractability we shall assume that there are only a small number of scenarios to be considered. In our problem a given scenario corresponds to a specific workload. The design team has the option of designing a system assuming that the most probable scenario will develop, or of designing systems for different scenarios and then studying the behavior of a system designed for one scenario assuming that some other scenario develops. The design group may also decide that the degree of uncertainty is too large to design a viable system and that more accurate data is needed. In this case the design group must have a rational scheme for trading off the additional cost of obtaining more accurate data with the improved design that will result from the better data. Bayesian decision theory suggests the following reasonable way of handling the problem.

All estimates regarding the future are subjective to some degree. It is possible (and indeed necessary) for a designer to assign an a priori probability to the occurrence of a given scenario. It is not satisfactory (and it has been argued that it is irrational) to restrict attention to the most likely scenario or to the worst case scenario. The network designed will depend upon the probability assigned to the different scenarios. Thus different designers may rationally make different design decisions because they have different estimates of the probabilities of occurrence of future scenarios.

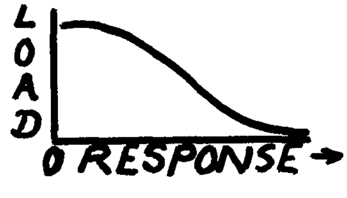
We shall assume that a workload is completely specified by the query and update rates generated at each branch and central office. If there are F functions and B branches, the workload (or scenario) may be specified by a 4xFxB - tuple : (update and query rates) x (branch and central offices) & (F functions) x (B branches). In theory any non-negative 4FB-tuple is a potential scenario; however computational tractability forces us to restrict attention to a small number.

Projecting demand for computing systems is especially difficult since the demand is not independent of the speed with which the system responds. If the system responds rapidly to a given load, the load is likely to increase, whereas if the system responds very poorly to a given load the load is likely to decrease Figure 3 shows a possible load-response time characteristic. We shall assume that the system operates on the relatively flat portion of the load-response time curve and thus the dependence of load on response time will be ignored in this paper.

1.3.2 Estimating network cost

The total cost of the network is generally not precisely fixed; typically the organization is willing to spend more money on the system if in so doing the behavior of the system in terms of response time (for instance) improves substantially. The design team is required to obtain estimates of some system objective such as response time as a function of the total cost of the system figure 4.

LOAD - RESPONSE TIME
Figure 3



Management can choose a suitable expenditure by studying the cost - response time characteristic. In Bayesian terms (as will be discussed later) the utility deriving from the network in terms of profits or goodwill has to be maximized.

1.4 Design variables

The variables that we shall work with are listed below.

(1)The number of copies of, and locations for each data base. Each branch office data base for each specific function can be located either at the branch office itself or at the central office, independent of the location of all other data bases.

A branch office's data base cannot be stored at any other branch office. We shall assume that there is only one copy of each data base. The analysis is easily extended to multiple copies. If a data base pertaining to a given branch office is stored at the branch office, then transactions generated at the central office which have to access the data base will have to travel along the communication link, to the branch office, get processed and possibly the reply will have to come back; however transactions generated at the branch office itself result in no traffic along the communication link. Conversely, if the data base pertaining to a branch office is stored at the central site, then transactions generated at the branch office will have to travel along the communication link to the central office, whereas transactions at the central office itself do not utilize the communication lines.

(2)Bandwidths of communication lines between central and branch offices

High bandwidth communication lines result in small communication delay but are expensive if there is a great deal of traffic along communication links then high bandwidth lines are required to maintain satisfactory response times. The cost - bandwidth structure of communication lines often exhibits economy of scale.

(3)Processing power (bandwidths) of data processing systems at the central and branch offices.

We shall assume that the total characteristics of a system are considered in computing its bandwidth. Thus a system consisting of a CPU and a set of disks is considered to be different from another system consisting of the same CPU and some other I/O subsytem. More expensive data processing systems result in less computational delay.

1.5 The objective function of the design

An objective of the design could be to minimize the average response time of a transaction. With this objective function, a network in which one branch office receives a response time of (almost) 0 minutes and in which a similar branch office receives a response time of 1minute would be as good as one in which both branch offices receive 1/2 minute average response times. In most environments the latter network would be generally prefered. Bayesian decision theory suggests methods to formally explain this preference. The formalism is based on utility theory.

For each type of transaction (for instance all transactions which modify a given data base may be thought of as being of one type) We shall assign a utility to each response. Short responses will have greater utility than long time responses, see figure5 below.

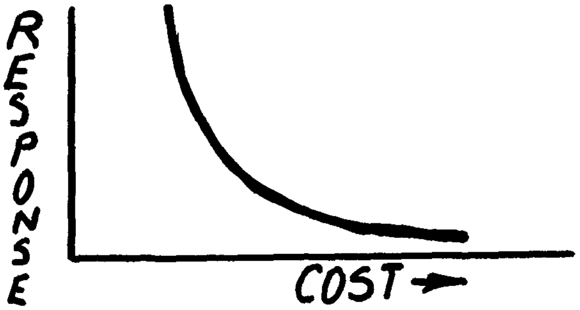


Figure 4

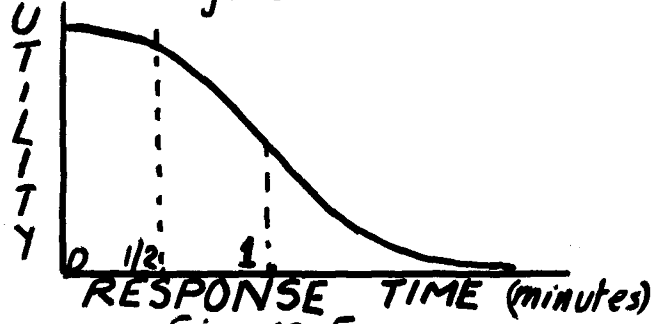


Figure 5

Typically the utility of a half minute response minus the utility of a 0 minute response is greater than the difference between a 1 minute and a half minute response. Thus a network in which two identical branch offices had identical mean response times of half a minute would tend to have higher utility than a network in which one branch office had a mean response time of 0 and the other had response times of 1 minute.

The distribution of response times (and not merely the mean value) affects the expected utility. Generally, given the same mean response time, distributions with smaller variance would tend to have higher utility than distributions with large variance. This is shown in figure 6. The preference for systems with smaller variance in response time also fits in with our intuitive expectation.

2. ANALYSIS

2.1 Definitions

In this paper we focus on decision-theoretic aspects of the design; we shall therefore assume that there exists a queueing model which derives the response time distribution from given system parameters.

Many different disciplines could be used to serve transactions at the central site. We shall assume that a certain proportion of the central processing site is reserved for transactions pertaining to a specific branch office. Let C be the capacity of the central computer in transactions processed per unit time and let g_b be the proportion of the central computer dedicated to the b-th branch office, $b = 1,..,B$ where B is the total number of branches, and let g_0 be the fraction of the central computer dedicated to processing transactions generated at the central site. For the purposes of computing response time distributions we shall assume that there are B+1 independent processors at the central site where the b-th processor has capacity g_b.C transactions per unit time and is dedicated to processing transactions by the b-th branch office for b = 1,..,B and by the central office for b=0. figure 7

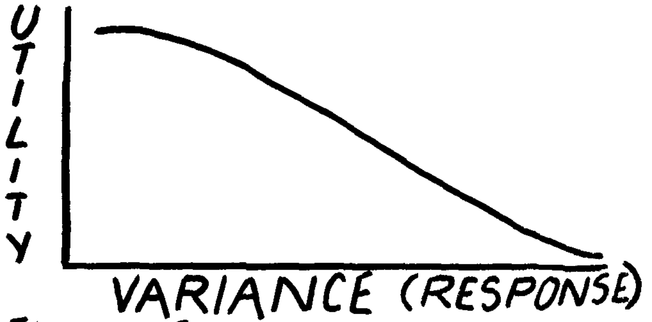


Figure 6:

We shall solve the problem by a dynamic programming technique, i.e. we shall optimize the total system by successively optimizing increasingly larger subsystems.

Let the system support F functions. Thus a given branch office may have F independent data bases. Each one of the F independent data bases may be stored either at the branch itself or at the central site. There are therefore 2^F possible schemes for storing the data bases pertaining to a given branch office. Since there are B branches there are a total of

$$2^{F^B}$$

different schemes for arranging the different data bases in the network.

Let there be L types of communication lines connecting branch and central offices, and M types of processors from which we must select one for each branch office and one for the central office. The number of possible configurations we need to consider is approximately

$$(2^F \mathrm{x} L \mathrm{x} M)^B$$

for each scenario. If F=L=M=4 and B=12 we have 2^{96} possible configurations to be analyzed for each scenario. This is intractaable and therefore better methods are required

Assume that the maximum processingpower of all the central site systems under consideration is C transactions per unit time; assume further that the units are defined to be small enough so that we need only consider allocation of whole numbers of units to each branch office (rather than fractional units) without significant loss of accuracy.

For each branch office b, we may consider a small number of possible values for

$$c_b = C.g_b$$

the amount of central processing power dedicated to the branch office. Assume that c_b is also integer. For instance, we may wish to choose the optimal values of c_b, while we know that the values should not be less than $LESS_b$ and should not exceed MAX_b. For each branch office we consider 2^FxLxMx(MAX_b-$LESS_b$) possible configurations for each scenario.

Let $U^i(b,a,l,m,c)$ be the expected utility of the transactions generated by the b -th branch office given data base allocation "a" (which is one of the 2^F possible allocations) communication line l between the branch office and the central office, system m at the branch office and capacity c of the central computer dedicated to the branch office, given scenario i. Let p_i be the probability assigned to scenario i.

Define

$$U(b,a,l,m,c) = p_i \times U^i(b,a,l,m,c)$$

2.2 The Algorithm

Compute U(b,a,l,m,c) for each possible b,a,l,m,c. If there are N scenarios the complexity of this computation is linear in N L,M,(MAX_b - $LESS_b$) and exponential in F.

U(b,a,l,m,c) is the utility to the b-th branch office deriving from the given configuration. Note that this value is dependent on the a priori probabilites p_i assigned to scenario i.

Let g(b,h,c) be the maximum utility to the b-th branch office given a total expenditure of h units on the processing system at the branch office and on the communication line connecting the branch office with the central site, and given that c units of the central site are dedicated to the b-th branch. Let $x_b(l)$ be the cost of the l-th type of communication line joining branch office b with the central site. Let $y_b(m)$ be the cost of the m-th type of system at the b-th branch office Then

g(b,h,c) = maximum of U(b,a,l,m,c) over all a,l,m such that $x_b(l)+y_b(m) \leq h$

The complexity of this computation is linear in h and c.

Compute g(b,h,c) for all b,h,c. Note that the cost h is assumed to be discrete and the units will have to be made large enough to result in computational tractability, while they are small enough to yield sufficient accuracy.

Let G(b,h,c) be the maximum sum of the utilities for branch offices 1,2,..,b, given that the total cost of all processors at branches 1,2,..,b and the cost of all communication lines linking branches 1,2,..,b to the central site do not exceed h, and given that there are c units of the processor at the central site dedicated to handling transactions from branches 1,2,..,b.

By definition;

G(1,h,c) = g(1,h,c) for all h,c and all c

It is easy to verify that

G(b,h,c) = maximum of G(b-1,h',c')+ g(b,h-h',c-c') over all possible h',c'.

We compute G(b,h,c) as b goes from 1 to B.

The complexity of this calculation is linear in B,H and C where H is the maximum cost of the system.

Let V(K) be the maximum total utility of the entire system as a function of the total cost K of the entire system. Let A(c) be the cost of the system at the central site as a function of its capacity c. Then

V(K) = maximum over all c of G(B,K-A(c),c)

This calculation is linear in H and C.

The function V(K) is the utility-cost function that management can use to obtain suitable designs.

An alternative way of picking a good design is to explicitly consider the utility of the money being spent on the system, rather than inspecting the V(K) curve. In this way we tradeoff the utility of the money being spent on the system (which would be used for something else if the system were not built) with the utility deriving from the system itself.

2.3 The cost of uncertainty

Management may wish to consider spending more time and money to obtain a more precise estimate of the load. Bayesian techniques provide a rational way of comparing the cost of obtaining more accurate data (perhaps by means of questionairres, or by studying simimar installations elsewhere,..) with the benefits that may accrue from better data. Though there exists a great deal of literature analyzing this tradeoff, we shall restrict attention to a relatively simple bound: the expected value of perfect information.

It is useful and interesting to compare th utilities of a system in which there is no uncertainty with systems in which there is uncertainty. Equivalently we wish to compare systems designed given just one scenario with systems designed given several scenarios.

Clearly, if we knew which one of the scenarios was going to develop, we could design a system specifically for that scenario and thus obtain greater utility. We shall now address the question: how much utility is lost due to uncertainty?

Let V(i,K) be the utility of the optimum network designed given that the i-th scenario will develop. We could compute V(i,K) in the same way as shown above, except that p_i=1.

Then the expected utility of the total system assuming that we know which scenario will develop in advance is: W(K)

$$\sum_i p_i . V(i,K) = W(K)$$

The difference W(K)-V(K) is the loss of utility due to uncertainty regarding the system. It is the upper bound on the utility to be gained by obtaining more accurate data.

It is more correct to define the value of perfect information in the following way: Let Z(i) be the maximum utility of a system designed given that the i-th scenario will develop; Z(i) is determined by optimally trading-off the utility derived from the network with the utility of the money spent on the system. Define $Z' = \sum_i p_i . Z(i)$

The difference Z'-Z, where Z is the maximum utility obtained from the network after considering cost given the probability distribution over scenarios, is the expected value of perfect information.

CONCLUSION

This paper has attempted to explore the use of Bayesian decision theory to handle uncertainties in the design process. Though a great deal of work has been done in the Bayesian Decision theory area, relatively little has been done on applications of the theory to computing systems. This is probably because the performance of computing systems is not well understood. However, Bayesian theory seems to have some potential to help in evaluating tradeoffs involving uncertainty.

REFERENCES.

Pratt,Raiffa and Schlaifer: Introduction to Statistical Decision Theory, McGraw Hill(1965).

ACKNOWLEDGEMENTS

The author would like to thank the State of Texas Department of Mental Health and Mental Retardation for suggesting the problem. He would also like to thank Professor James C. Browne of the University of Texas at Austin for his help. Thanks are also due to IBM T.J. Watson Research Center for the time spent on this problem.

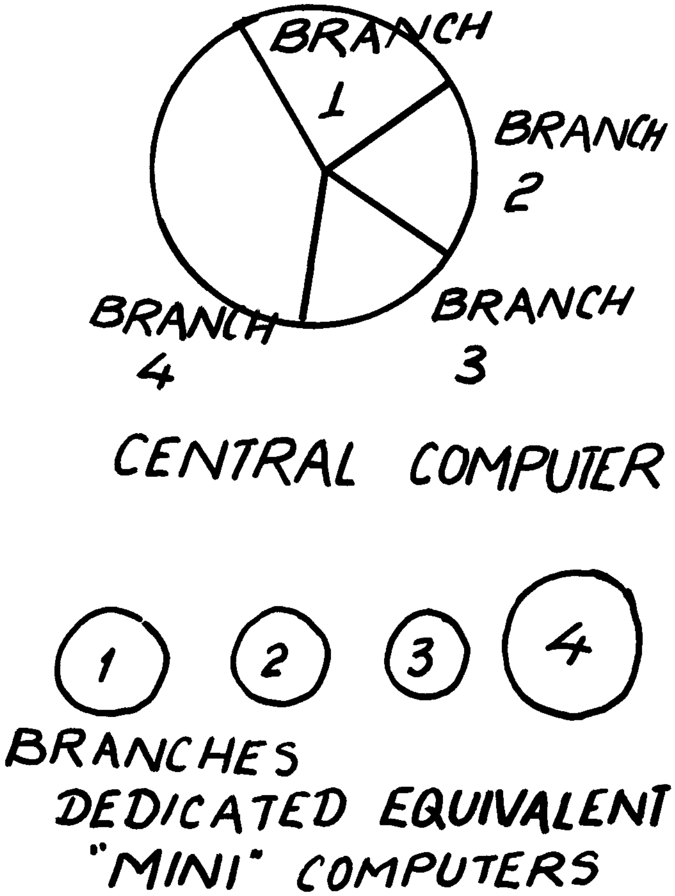


Figure 7