# THE APPLICATION OF SIMULATION IN COMPUTER SYSTEM DESIGN AND OPTIMIZATION

William C. Thompson
Vice President
Software Products Corporation
Falls Church, Virginia

As computer systems have grown in size, complexity, and cost, the emphasis in the computer field has turned from a "need for programmers" to a "need for systems analysts". There are almost as many definitions of systems analysis as there are analysts. However, for purposes of discussion I will define a systems analyst as a person who, given an automation objective, can design a computer system which, upon implementation, meets the objective in some way, e.g., minimum cost or maximum performance. Considering the cost of hardware today it is not without justification that "successful" systems analysts are highly paid technicians. Considering the age old economic law of supply and demand, it seems reasonable that this shortage would quickly be filled. However, this has not occurred because of the large experience requirement that systems analysts must have. There just are not that many "old timers" around to meet the demand.

Why the large experience factor? It is a simple fact that the computer business has expended all its efforts in helping to aid anyone else except its own. We in the computer field have produced little if any computer "tools" which can directly benefit the systems analyst. Thus, the analyst must rely on "past experience" to a large degree to perform his functions. This method of systems analysis is of necessity rendered obsolete because of the complexity of today's systems and the extreme shortage of skilled analyst manpower.

A system properly designed can greatly increase the overall efficiency of a computer operation. Given an improved systems design tool, installations could be providing an improved service with a smaller hardware investment. Although simulators have been in use for several years, they mainly were designed to aid in the computer system evaluation process with very limited capability in the design or optimization area. These programs have been limited to measuring equipment performance. What is needed today is a tool designed to aid the system analysts in the task of optimizing the systems design effort.

Illustration 1 lists what this tool must be capable of.

Such a tool has recently been developed. This tool, called CASE (Computer-Aided Systems Evaluation) makes maximum utilization of the systems analyst, large scale computer systems, and the application of simulation to achieve this objective.

The CASE program uses the CASE library of hardware and software characteristics as parameters in a generalized model of a computer hardware/software configuration. This model can then be used to simulate the configuration's performance in processing a given workload. Extensive statistics are gathered during the simulation which are analyzed, reduced, and presented in a number of reports.

A key capability of CASE is the program's automatic system design feature. Using this feature the program can, for any specific configuration, produce a system design for the processing of a workload specified in a machine independent form. Thus, a wide variety of configurations can be simulated in order to determine the best configuration for a fixed workload.

General controls can be specified to influence the system design. Through the use of these controls and successive simulations the optimum system design can be determined for a specific configuration. The CASE analysis reports are designed to provide the information needed to make the configuration and system design modifications required to lead to the desired evaluation objective.

Illustration 2 lists the components of a computer system evaluation using the CASE system.

The relationship of each of these major components of the CASE evaluation is shown in Illustration 3 with the connections and the general flow of information among them.

1. Definition of the System to be Evaluated. In any CASE evaluation the fundamental inputs of the system are the workload definition and the initial configuration.

The defined workload may be specified in whatever level of detail it is known. During the early design phases, general characteristics of files may be specified, and as the simulation proceeds and more information becomes available, then more data can be input for successive evaluations. In preparing the initial workload, the analyst will define the workload in terms of file definitions, run definitions, run sequence definitions, and general type of system to be designed (Batch, Multiprogramming, Multiprocessing, or Real Time). In all cases file and run data may be imputted in general form and CASE will, within the initial control constraints, make specific determinations as to "best" devices to be utilized. As configuration adjustments are made these "best" devices may change to optimize the configuration under study.

2. CASE Programs. The modular design of the CASE system allows simulations to be made quickly and to simulate very extensively and completely. The CASE Program is also easy to modify or extend to account for computer technology changes. These programs and the relationship between then will be explored later in some detail.

3. CASE Library. The library of manufacturer's hardware and software characteristics is an imput to a CASE simulation. The simulator extracts from the library, technical characteristics of the particular hardware and software to be evaluated. It is important to note that both hardware and software characteristics are included in the library.

4. System Optimization. The systems analyst prepares the initial configuration and/or design adjustments to reach system optimization. These adjustments will be made after analysis of the information provided by the CASE system. With each iteration and series of adjustments, the system will respond closer and closer to the desired result. These adjustments are of two types: design adjustments and configuration adjustments. Design adjustments are made after a number of preliminary iterations through the simulation. At this point, results are available which will show which are prime candidates among the given files and runs for consolidation, simplification, or elimination. Configuration adjustments may be made in conjunction with, or independent of, design changes.

5. Reports. Reports are provided at the completion of each simulation iteration and assist the analyst in determining the subsequent steps to take to get closer to the desired solution. They are also used to completely document the final solution in as great a detail as desired.

Illustration 4 lists the major reports produced by CASE. Many of these reports are the traditional outputs expected of an automatic timing program. However, some are specifically oriented toward "closing the design loop".

Illustration 5 is a sample from one such report. It shows information of the type essential to the design process - namely theoretical limits which could only be reached by sub-optimizing various components of the system.

The simulation system, CASE, is made up of the four major programs shown in Illustration 6.

The relationship of each of the CASE programs is shown in Illustration 7 with their processing and data flow connections.

The Independent Processing Analyzer (IPA) program performs the function of analyzing the control input, the initial configuration and workload, and simulating the system in a single run, or batch, mode of operation. In this mode of operation, the run to be simulated may utilize the entire configuration specified. The reports as a result of IPA show the statistics in regard to file requirements and utilizations. Also reported are configuration statistics which show utilization of each component in the input configuration.

The Concurrent Processing Analyzer (CPA) takes the results from the other analyzers plus configuration enhancements on an optional basis, then determines a run schedule and simulates the workload in a multi-programming mode of operation. This simulation shows the various run priorities and gives a time-period by time-period analysis of activity on all critical modules in the system under study.

The Real Time Analyzer (RTA) performs a very detailed simulation of real-time workload and is used only when this activity is present. The processing procedure, configuration facilities description, and the algorithms and logic of RTA constitute a model of the actual system. RTA simulates the initiation and flow of the real time activity through the system and records all pertinent statistics. The statistics gathered are later summarized and presented in reports which provide such information as response time, maximum queue sizes, and configuration utilization.

Time Sharing Analyzer (TSA) is a specialized program which analyzes performance characteristics of complex time sharing systems employing advanced techniques such as virtual memory, partial execution, and page swapping. TSA performs a detailed analysis of these systems in order to measure system performance characteristics which are configuration and workload dependent, and are otherwise unavailable.

In summary, CASE contains generalized models capable of simulating data processing systems of the current generation. Inherent in the design of the CASE system is the "man in the loop" which is a vital step in making this application of simulation a more meaningful tool for design and optimization.

It must be admitted that we still rely heavily on the systems analyst and on his skill and experience. In fact such tools as CASE provide another area in which the really competent and up-to-date systems analyst should be experienced. However, it must also be said that the capability of experienced analysts can be greatly amplified through proper use of this type of simulation tool.

1. Accurate simulation of modern computer systems.
2. Automatic capability for routine design decisions, thus simplifying input data preparation.
3. Fast response (turn-around).
4. Outputing comprehensive reports aimed at providing the key information needed to make design decisions.
5. Built-in control facilities for making design and configuration changes simply and without changing the basic input data.

Illustration 1

REQUIRED CAPABILITY

1. Definition of system to be evaluated
2. CASE Programs
3. CASE Library
4. System Optimization
5. Reports

Illustration 2

CASE System Components

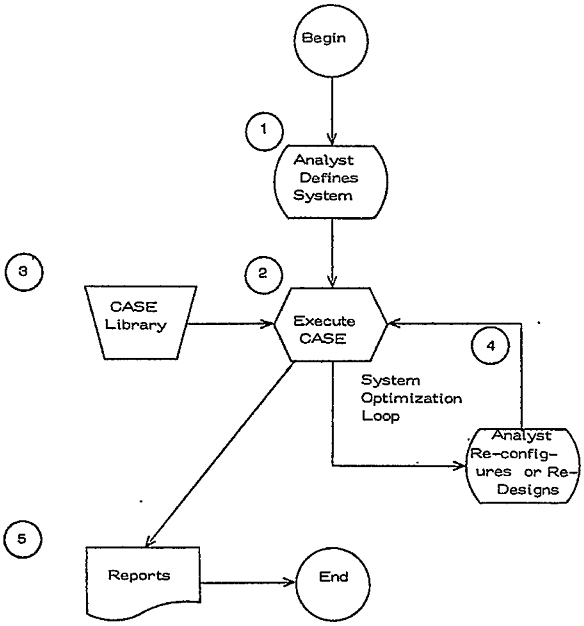


Illustration 3

CASE SYSTEM FLOW

1. File Description Report
2. Run Description Report
3. Configuration Report
4. Master Files
5. Run Detail Report
6. File Summary Report
7. Run Summary Report
8. Unit Utilization Report
9. Channel Utilization Report
10. Summary Report
11. Concurrent Processing Run Report
12. Concurrent Processing Slice Report
13. Concurrent Processing Run Analysis
14. Concurrent Processing Summary Analysis Report
15. Concurrent Processing Summary Report
16. Real Time Activity Report
17. Real Time Facility Utilization Report
18. Real Time Queue Analysis Report
19. Real Time Procedure Analysis Report
20. Real Time Event Report

Illustration 4

LIST OF REPORTS

OPTIMUM COMPONENT VALUE

| Size | | | | NUMBER | | PERFORMANCE | | |
|---|---|---|---|---|---|---|---|---|
| Memory | MT. | CR. | PR. | CP. | Channels | Processor | DF. | DP. |
| 69339 | 2.6 | 1.0 | 0.3 | 0.0 | 1.0 | 0.33 | 0.71 | 0.11 |

OPTIMUM COMPONENT TIME

| Memory | MT. | CR. | PR. | CP. | Channels | Processor | DF. | DP. |
|---|---|---|---|---|---|---|---|---|
| 213.4 | 156.0 | 285.4 | 165.3 | 0.0 | 194.0 | 198.2 | 425.9 | 68.9 |

CONCURRENT PROCESSING SUMMARY
ANALYSIS REPORT

Illustration 5

Independent Processing Analyzer (IPA)

Concurrent Processing Analyzer (CPA)

Real Time Analyzer (RTA)

Time Sharing Analyzer (TSA)

Illustration 6

CASE PROGRAMS

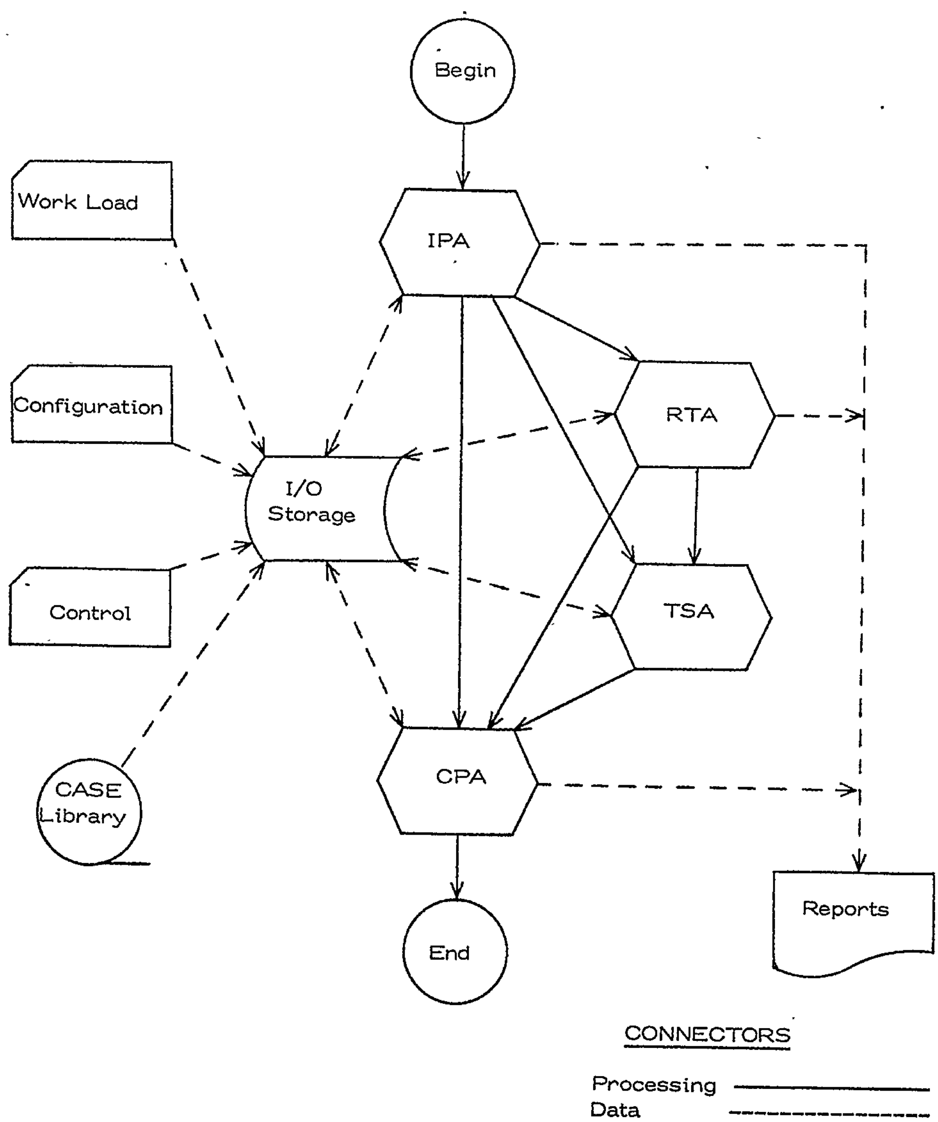


Illustration 7

CASE FLOW