

STATISTICAL CONSIDERATIONS IN SIMULATION
ON A NETWORK OF MICROCOMPUTERS

William E. Biles, Ph.D., P.E.
Industrial Engineering Department
Louisiana State University
Baton Rouge, LA 70803

Cheryl M. Daniels
Arthur Young Company
2121 San Jacinto St.
Dallas, TX 75201

Tamilea J. O'Donnell
Rockwell International
3370 Miraloma Avenue
Anaheim, CA 92803

ABSTRACT

Statistical considerations in simulating complex systems on a hierarchical network of low-cost microcomputers are discussed. Network configurations, allocation of computer tasks to nodes of the network, and assignment of simulation trials are examined. The binary tree and its X-tree variant are found to be especially attractive network configurations in simulation applications with simulation tasks assigned to nodes according to a model-frame/experiment-frame/output-frame trichotomy. Procedures for performing designed simulation experiments, including factor screening experiments, and for conducting variance reduction through manipulation of random number streams are presented.

INTRODUCTION

Computer simulations sometimes require exceedingly long computer time and very large addressable memory to execute realistic models of complex systems. Compounding this difficulty is the need to perform statistical and optimization studies of the model, which means that the model must be run repeatedly at varying conditions to evaluate the effects of selected input variables on the performance of the system. Much research has been devoted to designed experiments, factor screening experiments, principles of seeding, and other important issues in simulation experimentation. The impact of all this work on experimentation with complex computer simulation models is that such a project could require prohibitive amounts of computer time and demand a considerable share of available computer resources, and yet only provide meaningful results at the very end of an arduous experimental process.

The availability of low-cost microcomputers, and the capability to network them, gives us a much more powerful and efficient approach to this type of problem. What is needed is a systematic means of assigning computer experiments to a network of microcomputers in such a way that the statistical and optimization studies are vested with selected microcomputers while others are performing various assigned simulation tasks. There are many ways this system of work can be carried out, but this paper presents one way which takes advantage of a natural trichotomy prevalent in simulation modeling, that of establishing separate model, experimental and output subsystems within the simulation framework.

CONFIGURATIONS OF NETWORKS

There are several natural structures in which low-cost microcomputers can be arranged. For one, we can distinguish between logical structures and physical structures. Logical structures arrange similar processors into configurations which allow certain kinds of functions to be managed most efficiently. Physi-

cal structures are those in which the components of a processor are arranged for the most efficient operation of that processor itself. Thus, physical structures can be embedded within logical structures.

Harrison [7] describes three principal logical structures, as illustrated in Figure 1:

• Master-Slave

One computer is given the responsibility of overseeing and directing the activities of all other computers (slaves) in the network. The master computer assigns the tasks, provides load-sharing, and coordinates control of the subordinate computers.

• Hierarchical

A multi-level master-slave structure becomes a hierarchical configuration. In this arrangement, processors in each level have varying degrees of responsibility, with the degree of responsibility increasing at each higher level in the hierarchy.

• Peer-Connected

When no computer possesses power or precedence over another computer, they are said to be peer connected. In this case, the operating system must provide some type of scheduling algorithm by which work is distributed among the various processors.

Another method of classifying computer networks is the physical or communications structure of the network. Depain and Patterson [4] distinguish among five different types of physical structures in multi-processor organization:

• Multi-processor bus with shared memory

• Crossbar switch

• n-Dimensional cube

• Nearest neighbor connected arrays

• Cluster-bus

Figure 2 gives a schematic diagram of each of these organizations. Deitel [3] gives a detailed comparison of the features, advantages and disadvantages of these physical configurations.

One of the most appealing structures from the standpoint of computer simulation is the hierarchical configuration, especially the binary tree configuration illustrated in Figure 3. The advantage of this structure for computer simulation is that it allows the upper levels in the tree to be assigned the statistical, optimization, and executive functions associated with simulation, while allowing the more highly popu-

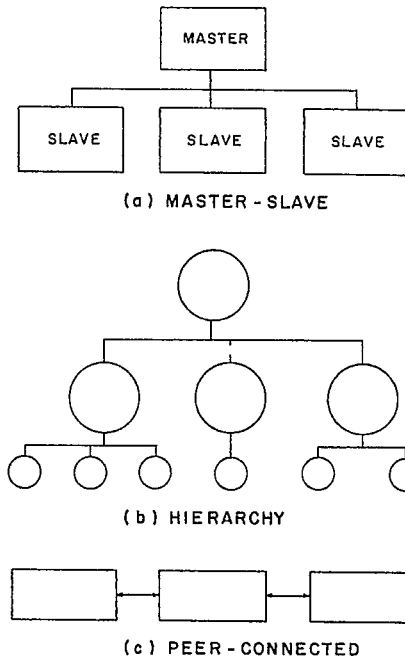


FIGURE 1. LOGICAL STRUCTURES IN MICROCOMPUTER NETWORKS

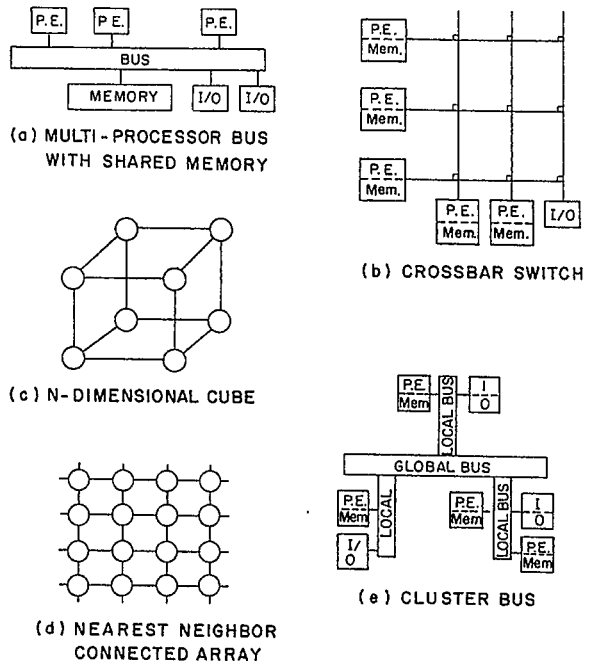
lated lower levels to actually perform simulation trials under varying input conditions. For this structure to function optimally, it would be useful to have a systematic mechanism for assigning models or sub-models, or specific sets of experiments, to given levels of nodes in the tree. Rozenblit [12] describes a top-down decomposition of the experimental frames in simulation that has elements that will be exploited here.

ASSIGNING COMPUTER SIMULATION TASKS

Various schemes have been put forth for assigning computer tasks to the nodes in a hierarchical network. Comfort and Miller [2] proposed task-dependent partitioning, in which functions are identified that are common to most computer simulations and these functions are assigned to specific processors in the network. The authors divided simulation tasks into five main categories:

- Event set processing
- Random number generation
- General simulation executive overhead
- Task dependent computations
- Input/Output processing

Comfort and Miller [2] found by analyzing a number of large simulation runs that event-set processing was performed 35 percent of the time, random number generation 3 percent, general simulation overhead 42 percent, and task-dependent computations 20 percent (I/O was ignored). The authors singled out event-set partitioning to be performed independently, setting up Motorola



Notes: P.E. is a processor
Mem. is a memory device

FIGURE 2. PHYSICAL STRUCTURES IN MICROCOMPUTER NETWORKS

M68000 processors to host specific sets of events. They were able to reduce waiting time in queue for events to be processed to only 8 percent of the single-processor mode of operation. Thus, multi-processor configurations based on event-set partitioning offer one, highly efficient means of gaining time savings in complex simulations.

Yet another concept for partitioning simulation rests on the system concepts of Oren and Ziegler [10]. These authors decompose simulation into six primary elements:

- Model structure
- Model outputs
- Input scheduling
- Initialization of the simulation
- Termination of the simulation
- Collection of simulated data

Oren and Ziegler go on to formalize a trichotomous approach to simulation modeling which consists of establishing three distinct "frames" - a model frame, an experimental frame, and an output frame.

Pegden went further, in developing the SIMAN simulation language [11], by formally incorporating the model frame/experimental-frame/output-frame trichotomy into the SIMAN program structure. Figure 4 shows the relationship of these three frames in the execution of a SIMAN model of a system. The essential aspect of simulation experimentation following a SIMAN-like modeling approach is that real-world systems possessing different physical configurations, or systems possess-

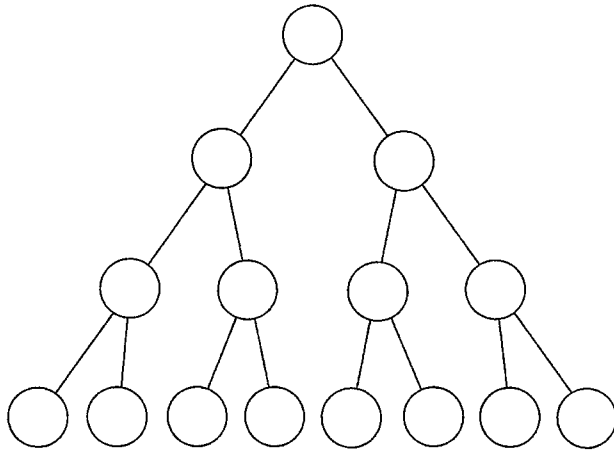


FIGURE 3. BINARY TREE NETWORK OF MICROCOMPUTERS

ing different physical functions, must be "remodeled" by altering the model frame. However, when the modeler desires only to evaluate the behaviour of a system under different operating conditions, this can be managed by simply providing different data sets in the experimental frame and leaving the model frame unaltered. The output frame, generated by a simulation trial at a particular combination of model and experiment frames, can be coupled with selected statistical and optimization software modules to analyze the output generated by several simulation trials. This feature of SIMAN will be invoked in proposing a systematic approach to assigning simulation tasks to a binary tree or X-tree network of low-cost microcomputers.

A NETWORK SIMULATION CONCEPT

The simulation approach proposed here is founded on the following features:

- Binary Tree or X-Tree Networks
- Model-Frame/Experiment-Frame/Output-Frame Simulation Structure

The binary tree network configuration was described earlier and is depicted in Figure 3. The X-tree is a minor logical variant of the binary tree in which there also exists peer communication with the sibling nodes to the left and right of a given node. Thus, a node beneath the topmost or "principal" node in an X-tree possesses the capability to communicate with the five following nodes:

- Parent node - immediately above in the network
- Children nodes - the two nodes immediately beneath
- Sibling nodes - the two nodes immediately left and right.

The X-tree is illustrated in Figure 5.

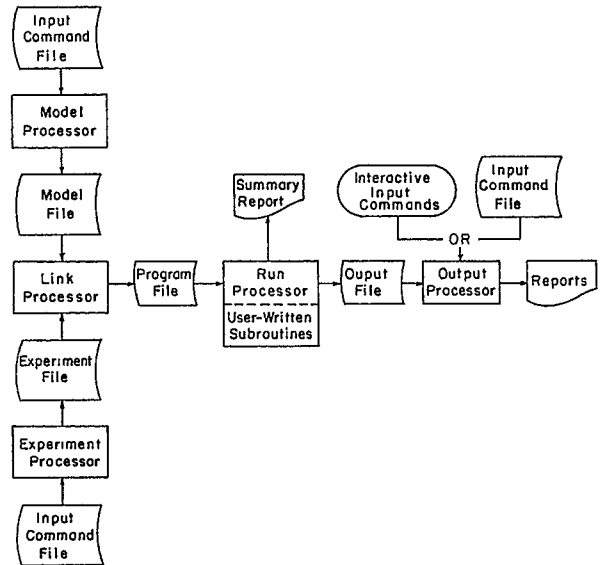


FIGURE 4. MODEL FRAME/EXPERIMENT FRAME/OUTPUT FRAME REPRESENTATION OF A COMPUTER SIMULATION

The second feature upon which the proposed concept rests is that of the model-frame/experimental-frame/output-frame trichotomy. In general, the several distinct model frames and their associated experimental frames will occupy the lower levels of the tree, with statistical and optimization functions accomplished at the upper levels. Executive control, including automatic assignment of particular combinations of model and experiment frames to nodes in the tree, selection of specific seeding rules for variance reduction, and the application of rules for the routing of files and data from one node in the network to another, is performed by the principal node which is situated at the apex of the tree.

For example, consider the case of a SIMAN simulation model of automobile manufacturing. Fernandes [6] desired to evaluate two different configurations of a Japanese automobile manufacturing system, and the two corresponding variations of a U.S. automobile manufacturing system. Because the U.S. and Japanese systems were substantially different, a total of four model frames were required to achieve the desired comparison. Moreover, suppose that three sets of experimental conditions were to be evaluated for each different model frame. One way to organize these simulation tasks within a binary tree network is shown in Figure 6. The tree requires four levels, totaling 15 separate nodes or microcomputers. Because each of the different experimental frames requires an entire SIMAN model, each of the four nodes on level 3 and the eight nodes on level 4 would need a 16-bit microprocessor with at least 256K of addressable memory to accommodate a SIMAN model frame with its associated experimental frame. Thus, the function of the two nodes on level 2 is to execute the statistical analysis of the output frames generated by the model/experiment frames simulated on the nodes beneath. Node 1 (level 1) is used for the overall optimization of the system, as well as any executive functions needed for the simulation study. The following section describes how simulation trials can be assigned to the nodes (microcomputers in a binary tree or X-tree).

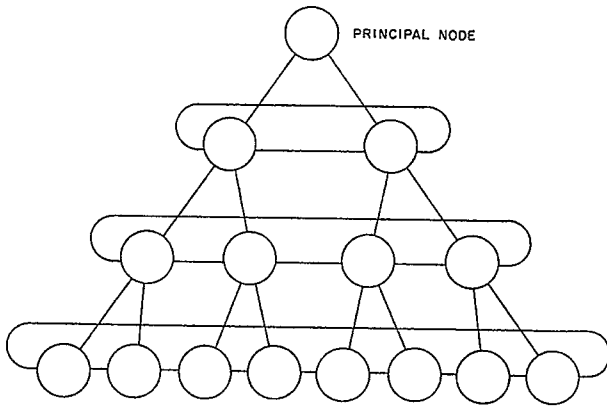


FIGURE 5. X-TREE NETWORK OF MICROCOMPUTERS

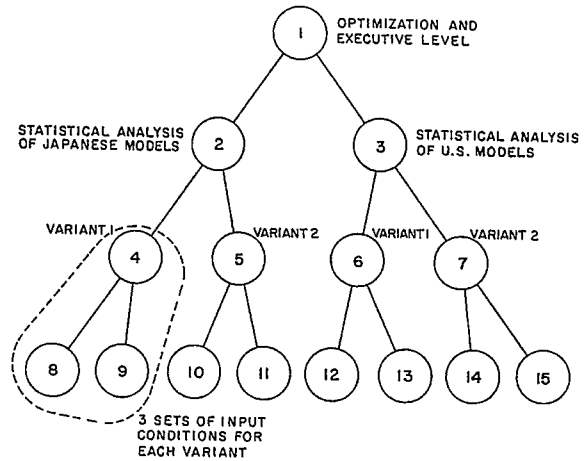


FIGURE 6. SIMULATION OF JAPANESE VERSUS U.S. AUTOMOBILE MANUFACTURING ON A BINARY TREE NETWORK

EXPERIMENTATION ON A BINARY TREE

Consider the binary tree shown in Figure 3. The primary node is denoted as node 1, and is located on level 1. Lower levels are numbered from 2 to m, and nodes are numbered from 1 to M, where

$$M = 2^m - 1 \quad (1)$$

For example, the number of nodes in a binary tree having levels 1, 2 and 3 would be 7.

The number of different sets of experimental conditions that can be evaluated simultaneously in a binary tree having m levels is M in equation (1) above. Therefore, our task is to define a binary subtree within the binary tree which possesses the requisite number of nodes on which to place the needed simulation trials.

Table 1 below gives a summary of the number of levels and nodes available in binary trees or subtrees. It remains to allocate the needed experimental trials to the appropriate subtree and perform the simulation trials there. It should be noted that each node in the subtree will execute the same model frame, but with a unique experimental frame. For instance, suppose we have three input variables $x_i, i=1, \dots, 3$ for which we must propose a designed experiment. If we wish to employ a 2^3 factorial design, we need 8 nodes. As seen in Table 1, we must use levels 1 through 4 for a total of 15 nodes, with the uppermost nodes serving as statistical and optimization functions. If, however, we are willing to eliminate one of the 8 design points, we can easily fit the 7 remaining simulation trials on a subtree possessing exactly 7 nodes. If we choose to delete the design point which gives the three-factor interaction effect (which we would very likely have difficulty explaining in any case) we would be able to gain the requisite information about the system being modeled with a minimum number of simulation trials, and hence employ the minimum number of microcomputers.

Table 1
Microcomputer Node Availability
in a Binary Tree

Level, k	No. of Nodes on Level k, 2^{k-1}	Total Nodes, $2^k - 1$
1	1	1
2	2	3
3	4	7
4	8	15
5	16	31
6	32	63

Note: the typical configuration of each processor node in such a network would be a 16-bit CPU, 256K of addressable memory, and a separate processor to route communications to other nodes and peripherals.

VARIANCE REDUCTION

One of the important statistical issues in simulation is that of variance reduction. This is achieved through appropriate selection of the random number stream associated with the simulation. The simulation model will contain a suitable random number generator which will be common to all experimental frames as well. Unique random number streams can be obtained through the choice of the initial random number seed. There are three approaches to variance reduction through random number stream control:

- Randomly selected random number streams at each experimental point, with the only stipulation that a given stream is not repeated at any other experimental point.
- Common random number streams, so that each experimental point sees precisely the same series of random numbers.

- Antithetic streams, in which one experimental point sees the stream $R = (r_1, r_2, \dots)$ while another experimental point sees the stream $\bar{R} = (1-r_1, 1-r_2, \dots)$. Since R is uniformly distributed in the interval $0 < r < 1$, \bar{R} is likewise uniformly distributed in the same interval.

These three alternative seeding schemes have been observed to have differing effects on the correlations between responses. When two sets of random number streams are generated using different randomly selected vectors of seeds, the resulting time series, and hence their means, are standardly assumed to be uncorrelated. The induction of positive correlation, and hence an increase in variance, has been substantiated by both analytical and empirical investigations (see Kleijnen [8]). The use of antithetic variates generates negatively correlated time series, and hence achieves the desired variance reduction.

Although this paper does not propose a specific choice of seeding mechanism, we do point out that Schruben and Margolin [13] have shown that antithetic variates are especially useful when conducting a designed experiment with a simulation model. These authors suggest that a designed experiment be broken into two blocks, with block 1 using the set of streams R_1, R_2, \dots, R_k , where there are k experimental points in each block. Block 2 would then utilize the corresponding antithetic streams $\bar{R}_1, \bar{R}_2, \dots, \bar{R}_k$. Kleijnen [9] discusses the formation of blocks in simulation experiments. Wilson [16] also discusses the relationship of antithetic variates to multivariable simulation inputs.

Perhaps the best way to utilize variance reduction via antithetic variates when simulating on a hierarchical network of microcomputers is to conduct successive runs on the same processor with the antithetic random number streams, since both the model frame and the experimental frame are exactly the same from the first run to the second. Only the random number streams (R and $1-R$) will be different.

OPTIMIZATION EXPERIMENTS

The optimization of a simulated system is approached by employing response surface designs in the same manner described by Biles and Swain [1]. The requisite number of experimental points, each with its selected random number stream, are assigned to a subtree of the network. A higher-level node is employed to perform the mathematical analysis associated with the optimization. Biles and Swain describe several optimization approaches which can be taken in a complex simulation experiment. Any of these approaches can be adapted to the binary tree network of microcomputers. The concepts of Fedorov [5] are applicable to selecting values of the input variables $X(x_i, i=1, \dots, n)$ so that simulation experiments can be deployed optimally.

FACTOR SCREENING EXPERIMENTS

A precursor to optimization studies in simulation is that of factor-screening studies. Given a set of n input variables which are hypothesized to exert an influence on the simulation output, we seek to isolate the k most important factors. Smith and Mauro [14] describe several types of experimental designs which can be used in carrying out factor screening experiments. These experiments can be assigned to the lower-level nodes in a binary tree or X-tree configuration of microcomputers, with upper-level nodes used to perform the requisite statistical tests used to select

the most influential input variables. Then a second phase of experiments can be conducted to carry out the optimization studies.

SUMMARY AND CONCLUSIONS

Distributed simulation is a rapidly evolving area of computer technology. With hierarchical networks of low-cost microcomputers a modern reality, strategies for conducting complex simulations on such networks are essential for continued development in this field. This paper has described a technique that utilizes commercially available, low-cost microcomputer hardware and software arranged in binary trees. Other structures are feasible and, for some simulations, preferred. For instance, in astro- or plasma physics simulations a $4 \times 4 \times 4$ cubic grid of 64 processors can be employed, with each processor assigned a specific grid point and passing results to each other processor in the cube.

The present discussion assumes that each node in a binary tree possesses an entire model frame. It would also be feasible to partition a complex model into subsets, with data communicated between nodes as necessary during progress of the simulation. The automobile manufacturing models by Fernandes [6] exploited this capability.

REFERENCES

1. Biles, W.E. and J.J. Swain, Optimization and Industrial Experimentation, John Wiley and Sons, New York, 1980.
2. Comfort, J., and A. Miller, "Considerations in the Design of a Multiprocessor-Based Simulation Computer," in Modeling and Simulation on Microcomputers (L. Leventhal, Editor), Society of Computer Simulation, Lajolla, CA, 1981.
3. Deitel, H.M., An Introduction to Operating Systems, Addison-Wesley, Reading, MA, 1981.
4. Despain, A.M., and D. A. Patterson, "X-Tree: A Tree Structured Multiprocessor Computer Architecture," Computer Science Division, University of California, Berkeley, CA, 1981.
5. Fedorov, V.V., Theory of Optimal Experiments, Academic Press, New York, 1972, pp 211-220.
6. Fernandes, C., "A Computer Simulation Model to Analyze Production Processes in the American and Japanese Automobile Industries," unpublished MSIE thesis, Industrial Engineering Department, Louisiana State University, Baton Rouge, LA, December 1984.
7. Harrison, T.J., "Minis, Micros, and Miscellany," Chapter 8 in Minicomputers in Industrial Control, (T.J. Harrison, Editor), Prentice-Hall, Englewood Cliffs, NJ, 1978.
8. Kleijnen, J.P.C., Statistical Techniques in Simulation, Part I, Marcel Dekker, New York, 1974, pp 186-200.
9. Kleijnen, J.P.C., Statistical Techniques in Simulation, Part II, Marcel Dekker, New York, 1974, pp 287-407.
10. Oren, T.I., and B.F. Ziegler, "Concepts for Advanced Simulation Methodologies," Simulation, March 1979.

11. Pegden, C.D., Introduction to SIMAN, Systems Modeling Corp., State College, PA, 1982.
12. Rozenblit, J.W., "Experimental Frames for Distributed Simulation Architectures," Distributed Simulation, Simulation Councils, Inc., LaJolla, CA, January 1985, pp 14-20.
13. Schruben, L.W., and B. H. Margolin, "Pseudo-Random Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments," Journal of the American Statistical Association, Vol. 73, 1978, pp 504-540.
14. Smith, D.E. and C.A. Mauro, "Factor Screening in Computer Simulation," Simulation, February 1982, pp 49-54.
15. Wilson, J.R., "Antithetic Sampling with Multivariate Inputs," American Journal of Mathematical and Management Sciences, Vol. 3, No. 2, 1983, pp 121-144.

BIOGRAPHICAL SKETCHES

WILLIAM E. BILES

Dr. William E. Biles is Professor and Chairman of Industrial Engineering at Louisiana State University. A native of Mobile, Alabama, he received the BS degree in Chemical Engineering from Auburn University in 1960, the MSE degree in Industrial Engineering from the University of Alabama in Huntsville in 1969, and the PhD degree in Industrial Engineering and Operations Research from Virginia Polytechnic Institute and State University in 1971. Dr. Biles served on the faculties at Notre Dame and Penn State before assuming his present position. He is a member of the Engineering Accreditation Commission and Director for Research of the Institute of Industrial Engineers.

Dr. William E. Biles, Chairman
Industrial Engineering Department
Louisiana State University
Baton Rouge, LA 70803
(504) 388-5112

CHERYL M. DANIELS

Ms. Cheryl M. Daniels is a systems analyst with the Arthur Young Company in Dallas, Texas. A native of Pineville, Louisiana, she received the BS degree in Industrial Engineering from Louisiana State University in 1982. After working for one year as an Industrial Engineer at Western Electric Company in Shreveport, Louisiana, Cheryl returned to LSU to undertake graduate studies in Industrial Engineering. She received the MSIE degree in 1984 and immediately assumed her present position at Arthur Young.

Cheryl M. Daniels
Arthur Young Company
2121 San Jacinto St.
Dallas, TX 75201

TAMILEA J. O'DONNELL

Mrs. Tamilea J. O'Donnell is a software engineer for Rockwell International Corporation in Anaheim, California. A native of Baton Rouge, Louisiana, Jann received the BS degree in Industrial Engineering from Louisiana State University in 1983. She received the MSIE degree in 1984 and immediately joined Rockwell International in her present position.

Tamilea J. O'Donnell
Rockwell International
3370 Miraloma Avenue
Anaheim, CA 92803