Activity-based simulation offers functionality that we've never experienced until now

New tool for business process re-engineering

By Kofi Nyamekye

Imagine you are the plant manager of a large global manufacturing enterprise. You are called into the CEO's office and he tells you he just returned from a seminar on business process re-engineering.

"Our strategic vision for the next five years is to cut all operational costs by 50 percent through re-engineering our processes," says the CEO.

Is this corporate vision plausible? How will you proceed? Such a vision is not unreachable; in fact, a powerful new system will help you achieve it.

Let's begin with a definition or two. A business process is a sequence of two or more activities that serve a purpose for an enterprise. For example, an injection molding process consists of clamping the die cavity, injecting the liquid polymer, compressing the polymer, allowing the polymer to cool, and ejecting the part from the cavity.

As defined in *Reengineering the Corporation*, business process re-engineering is "the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical measures of performance, such as cost, quality, service, and speed." This is not a new idea. In fact, it dates back to at least a century ago, with roots in the work of Andrew Carnegie and Frederick W. Taylor.

The foundation for BPR

Many companies use activity-based costing as the foundation for BPR. The ABC concept is that every enterprise consists of resources, activities, and cost objects (see Figure 1). Activities are defined by decomposing each business process into individual tasks. Then the cost of all resources consumed by each activity and the cost of all activities consumed by each product or cost object are tracked. (Note that activities can also be consumed by other activities.) If we re-engineer an activity, the cost of all resources consumed by that activity and the cost of that activity consumed by each cost object are reduced. Consequently, the fundamental concept of ABC is the same as that of the early scientific management system.

While the founding fathers of scientific management focused on the detailed improvement of individual activities, time did not permit them to study the interaction among activities and their costs. For example, what happens to the overall efficiency of a production line if a machine in it fails? If the line produces certain quantities of products X, Y, and Z before the machine fails, how do we trace the maintenance costs to each product? How does the machine downtime affects other operators' activity times within the line? If through business process re-engineering we can eliminate the breakdown of this machine, what will be the new cost of each product? To date, no traditional activity-based costing system can capture the dynamics of an enterprise and map the cost flow from resources to activities and from the activities to cost objects.

A fundamental limitation of ABC systems deserves attention here. In many enterprise systems (except for continuous processing systems) resources are used at random intervals, and raw materials arrive at a process at unpredictable discrete time intervals. Direct and indirect labor is performed on these materials at random intervals. After activities are completed, the parts move to the next process, leaving the resources idle momentarily until the next batch of raw materials arrives. Because the resource usage is a stochastic (random) process, managers of an enterprise are challenged to estimate the true resource expenditure. For example, even if just 60 percent of a quality control inspector's effort is consumed, the resource expenditure (cost) for this resource is still based on 100 percent utilization. Consequently, managers usually base resource expenditures on 100 percent utilization of resources, not the true, partial utilization of the resources. The result is that significant errors are introduced into the budget.

Furthermore, because the enterprise does not usually know how to handle the unpredictable use of resources, significant inefficiencies exist, causing under-utilization of resources. Current ABC systems stumble over this major issue. They cannot predict the idle capacity; therefore, they cannot predict true resource expenditures.

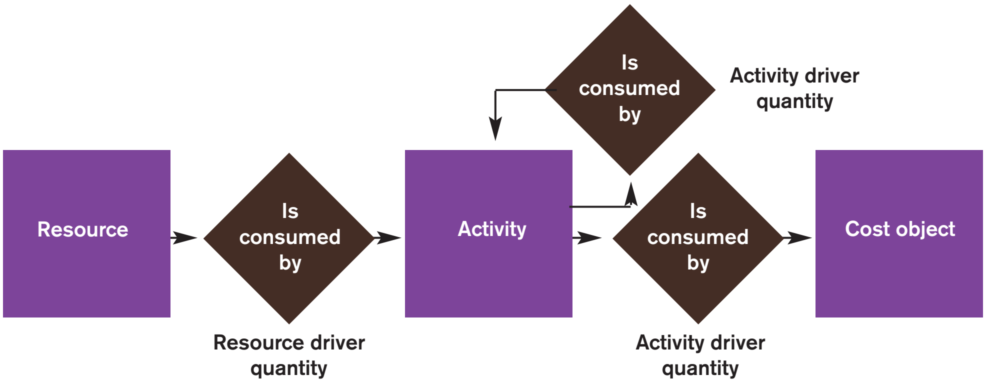
A recent breakthrough in activity-based simulation addresses all these issues.

When a machine fails and is down for some time before is repaired, the economic effect of the downtime and repairs is usually unknown from the simulation results.

The next tool for BPR

The resources of any enterprise consist of direct labor, direct materials, and overhead. Traditionally, manufacturing systems simulation models have focused attention only on resource utilization at work centers, with no emphasis on their economic impact. Usually, a prerequisite of any manufacturing systems simulation is to define the labor-related unit-level activities, such as drilling a hole or making a mold at a work center. The direct labor resource or a machine operator is then specified; the job arrival and departure times and the activity times are then defined. Occasionally, a job setup (a batch-related activity) may be specified. In this case the tool-and-die maker may be defined as the resource needed to do the machine setup. Machine maintenance (a technology-sustaining activity), which consumes the maintenance crew's efforts, may also be considered. Here, the mean time to failure and the mean time to repair are the parameters needed for simulating machine failures, downtimes, and repairs. Once these data are input, the simulation is allowed to run to see the system's behavior for a period of time. The performance measures typically used to judge the resource utilization at the work center are the percent time the direct labor is busy, the percent time the machine is busy, and the percent utilization of the maintenance and machine setup crews.

The weakness of the traditional modeling approach is that we cannot use the simulation results directly to find the



Resource driver quantity measures the rate at which the resource is consumed by the activity. Activity driver quantity measures the rate at which the activity/cost object is consumed by the activity/object.

Figure 1. An entity relationship diagram for modeling the activity-based database system.

cost of resources consumed by activities at work centers. When a machine fails and is down for some time before is repaired, the economic effect of the downtime and repairs is usually unknown from the simulation results. Consequently, managers do not know which activity costs the most and what radical improvement may be needed to reduce the cost. The lack of such capability limits the usefulness of the current simulation models in helping managers reduce product costs and increase product quality.

Other sources of variability that can significantly impact product cost in manufacturing environments are:

- fluctuations in activity time due to differences in machine operators, production equipment, and material;
- job setups;
- operator availability;
- defects and reworks; and
- tool-and-die failure.

Unless the sources of the variability are understood and attacked, no business process re-engineering system can succeed. In fact, even if a completely new process is designed to replace a system, no guarantee exists that the product cost from the new process will be reduced and the process capability will be improved. Defects from raw material suppliers can cause the new process to produce scrap, consequently increasing the price of all products. In addition, die failure can increase product cost significantly. Meaningful cost reductions are impossible to achieve without rooting out all sources of variability after new processes are in place.

Identifying and attacking sources of variability is a major challenge even after BPR is instituted in an enterprise.

Note that sources of variability exist not only in a manufacturing enterprise but in other enterprises as well. In the retail industry one of the major sources of variability is defective merchandise. Not only do defects increase inventory costs, they raise the original cost of the items by 100 percent. Thus, identifying and attacking sources of variability is a major challenge even after BPR is instituted in an enterprise. The new activity-based simulation system addresses these issues.

ABSS components

The primary components of an activity-based simulation system are:

- **Activity-based database system**. The activity-based database system is a set of related files in the activity-based simulation system. The records in one file may be related to the records in a different file; for example, a resource record may be linked to activity records. In turn, each of the activity records may be linked to other relevant activity records, and each of the activity records may be linked to cost object records. Figure 1 shows the database model known as the entity relationship diagram. A data entity, represented by a rectangular shape, can be anything real or abstract about which we are interested in storing data. Resource, activity, and cost objects are the three basic entities about which companies want to store data. The diamond shape represents the relationship, such as "is consumed," among the entities. Figure 2 shows an example of the resource data set from the ABSS system.

As noted before, many companies use their activity-based costing systems as the foundation for BPR and thus the design of their enterprise-wide systems. The design of an activity-based database system must be flexible enough to include other enterprise data such as the bill-of-materials, employee data, and so on. Any data repository such as Microsoft Excel can be used to build the data files for a prototype ABSS. After the prototype system has been successful, the data files can be migrated to a full-scale database such as Oracle or Microsoft Access.

Reference Number	Type of Level	Name	Account
PRD	1	PRODUCTION	C
PRD_DL11	2	Direct Labor Wage (R)	A
PRD_DL11_1	2	Direct Labor Wage (CE)	E
PRD_ST12	2	Tooling Setup Wage (R)	A
PRD_ST12-1	2	Tooling Setup Wage (CE)	E
PRD_CM13	2	Machine Repair Cost (R)	A
PRD_CM13-1	2	Conveyor Maintenance Wage (CE)	E
PRD_DT14	2	Maintenance Downtime Cost (R)	A
PRD_DT14-1	2	Conveyor Downtime Cost (CE)	E
PRD_CD15	2	Machine Depreciation Cost (R)	A
PRD_CD15-1	2	Conveyor Depreciation Cost (CE)	E

Figure 2. An example of a resource data set.

Rather than use the entity relationship diagram modeling approach for designing the database, object-relational data modeling extensions can be used to construct the activity-based database system. The object-relational modeling methodology provides much more flexibility than the entity relationship diagram approach.

- **Activity-based costing simulator**. The activity-based costing simulator models the detailed costs of resources consumed by activities and the cost of the activities consumed by the cost objects within an enterprise. For each business process, the simulator must find the cost of each activity and then roll up the total cost of the activities to each process level. The activity costs must include the costs due to the impact of the sources of variability on each process, for example the cost of defects and others. Figure 3 shows the partial activity costs from the ABSS. The cell level (Cell 1, Workcenter Activities) shows one process: loading base. The loading base process consists of the following activities: walk to LB station for setup (A); walk to LB station for load (A); set up base (A); and load base (A). Because three different parts (P1, P2, and P3) are processed by the cell operator (Figure 4) at the loading base process, three special activities have been designed to hold the costs for each part. The accumulators are: accumulate LB cost for P1 (A); accumulate LB cost for P2 (A); and accumulate LB cost for P3 (A). The accumulator costs ($14,800.35; $16,835.30; and $14,389.20) are added and rolled up to the loading base process level ($46,024.85).

Thus, with the detailed true activity costs, managers of the enterprise can examine in detail the true costs of business processes and determine which ones must be re-engineered. With the what-if scenarios, managers can use the ABSS to find alternative re-engineered processes before selecting the processes that meet corporate strategic goals.

Oros 99 SAP1 was the activity-based costing simulator used in this ABSS; any activity-based costing simulator can be used.

◆ **Activity-based process simulator.** The activity-based process simulator uses business data and sophisticated mathematical models to forecast implications of alternative decisions in an enterprise. A graphic model of the business processes in the sub-system or the whole enterprise is constructed. The processes are modeled in real time using business data. Statistical data about each process are collected. These data are known as the resource and activity driver quantities. The resource driver quantity measures the rate at which a resource is consumed by an activity. Similarly, the activity driver quantity measures the rate at which an activity is consumed by an activity or an activity is consumed by a cost object. Examples of the driver quantities are the number of setups, the number of defects, the number of parts, the number of repairs, and so on. These data are then exported into the activity-based system database. The activity-based costing simulator imports them and other cost data for the detailed cost analysis.

ProModel was the activity-based process simulator used in this ABSS; any activity-based process simulator can be employed to design the ABSS.

◆ **Graphical user interface.** A graphical user interface should be designed for each primary component; this allows users to communicate with any component of the ABSS easily.

Secondary or optional components that can be added to the ABSS later are:

◆ Material requirements planning, which uses the simulated lot size data from the database for ordering dependent-demand items.

◆ Conditioned-based maintenance simulator, which sends mean time to failure and mean time to repair data to the database, to be used by the process simulator.

◆ CAD tool

◆ Robot simulator

◆ End-user presentation tool (for displaying other pertinent data)

◆ Enterprise resource planning

◆ Intelligent enterprise-wide controller,

Name	ReferenceNumber	FY1 MTTF_30	DriverQuantity
Activity		$399,405.70	
CELL 1 WORKCENTER ACTIVITIES	CELL 1	$399,405.70	
Loading Base	LB2	$46,024.85	
Walk to LB_Station for Setup (A)	LB21	[illegible]	
Direct Labor Wage (R)	PRD_DL11	[illegible]	0.09
Walk to LB_Station for Load (A)	[illegible]	$109.05	
Direct Labor Wage (R)	PRD_DL11	$109.05	7.27
Setup Base (A)	LB23	$2,295.00	
Tooling Setup Wage (R)	PRD_ST12	$2,295.00	[illegible]
Load Base (A)	LB24	[illegible]	
Direct Labor Wage (R)	PRD_DL11	[illegible]	[illegible]
Accumulate LB Cost for P1 (A)	[illegible]	[illegible]	
Walk to LB_Station for Setup (A)	[illegible]	$3.45	0.23
Walk to LB_Station for Load (A)	[illegible]	$51.45	3.43
Setup Base (A)	LB23	$765.00	17.00
Load Base (A)	LB24	$97.45	[illegible]
Base	B_1	[illegible]	
Defects for Base	[illegible]	[illegible]	
Accumulate LB Cost for P2 (A)	[illegible]	[illegible]	
Walk to LB_Station for Setup (A)	[illegible]	$3.45	0.23
Walk to LB_Station for Load (A)	[illegible]	$27.75	1.85
Setup Base (A)	LB23	$765.00	17.00
Load Base (A)	LB24	$62.10	4.14
Base	B_1	[illegible]	
Defects for Base	[illegible]	[illegible]	
Accumulate LB Cost for P3 (A)	[illegible]	[illegible]	
Walk to LB_Station for Setup (A)	[illegible]	$3.45	0.23
Walk to LB_Station for Load (A)	LB22	$29.85	1.99
Setup Base (A)	LB23	$765.00	17.00
Load Base (A)	LB24	[illegible]	[illegible]
Base	B_1	[illegible]	
Defects for Base	[illegible]	[illegible]	

Figure 3. A partial data set of activity costs.

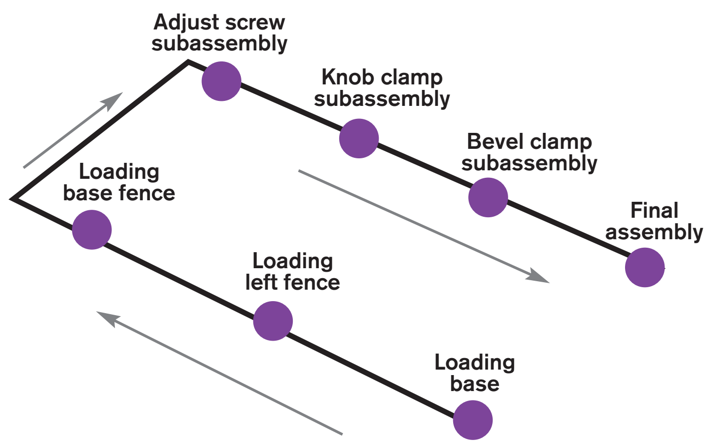


Figure 4. An assembly line prototype for a power tool manufacturer.

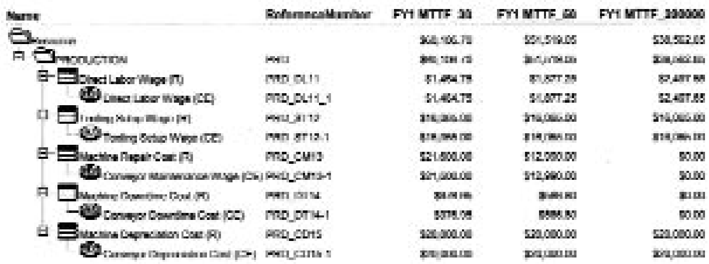


Figure 5. The resources consumed by the assembly line.

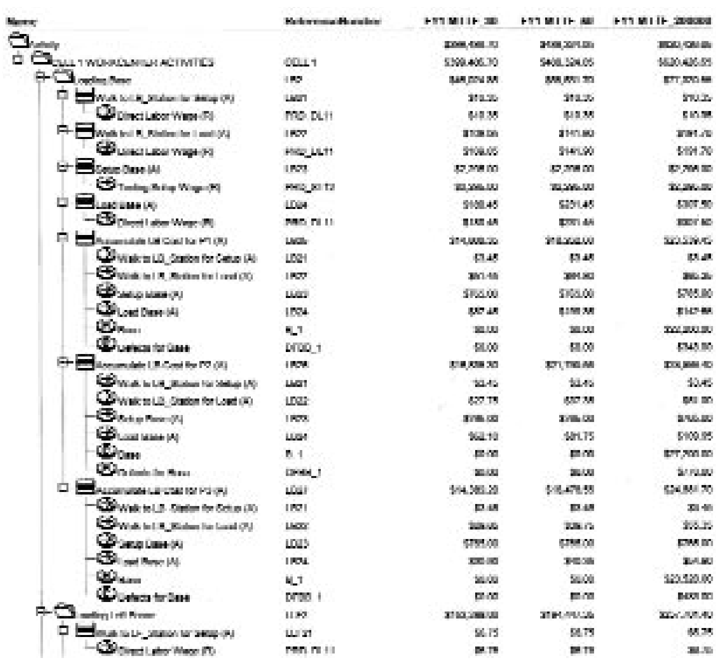


Figure 6. A partial detail of activity costs.

which analyzes the data to uncover problems hidden in the data relationships and alerts management for remedial actions.

A prototype ABSS

A prototype of the activity-based simulation system was constructed for one of the cellular assembly lines of a major power tool manufacturing company. One operator works the cell and a maintenance person repairs the U-shaped conveyor whenever it fails. Figure 4 shows the processes in the assembly line. Each process level is then broken down into detailed activities. The resources that will be consumed by each activity are identified. Similarly, the activities that will be consumed by other activities are identified, as are the activities consumed by the cost objects (P1, P2, and P3). Taking into account the resources, activities, and cost objects, the entity relationship diagram is used to construct the relational database.

ProModel then simulates the driver quantities for three different mean times to failure for the conveyor: 30 minutes, 50 minutes, and 200,000 minutes. The line is simulated for about three months. After each simulation run, ProModel populates the database with the driver quantities. The database also has the activity-based general ledger module that predicts the activity-based general ledger for each run. Unlike the traditional general ledger that summarizes expenses at the department level, an activity-based general ledger summarizes the overhead and direct labor expenses directly consumed by activities. A graphical user interface links ProModel and the data repository.

Oros 99 then imports the appropriate data modules for cost simulation. Figure 5 shows the total cost of the resources consumed by the assembly line; Figure 6 is a partial detail of activity costs; and Figure 7 shows the unit cost objects. Consider Figure 5: Each time the conveyor breaks down on the average of 30 minutes, the cell operator becomes idle. So, his or her true wage for the first quarter is $1,464.75 (based on wage of $15 per hour and a total of 97.65 hours for direct labor-related activities for nine weeks). Note that the same operator sets up the job at a different labor rate. However, the conveyor's repair crew wage is $21,600. If, through preventive maintenance, the conveyor no longer fails (because the mean time to failure is 200,000 minutes), the cell operator's wage is $2,497.65, which agrees with the accounting department's report. That is significantly better than $1,464.75. The maintenance crew's wage is $0. The numbers prove that this enterprise wastes labor resources when the conveyor breaks down frequently. Figure 7 also deserves attention. It shows that if the mean time to failure is eliminated completely, the unit cost is reduced by 9.24 percent. Other sources of variability, such as defects and setup time, can be changed to reduce the unit cost further. Operator availability could also be improved if the operator's traveling times are completely removed.

The interactions among resources within a system are important. When equipment breaks down, the maintenance crew is needed to make repairs. From a traditional accounting viewpoint, manage-

Bar-code scanners connected to a PC can be used to monitor inventory levels of each part for every process at a workstation.

ment is getting its dollars worth by keeping the repair crew busy. However, while the equipment is waiting to be fixed and during its repair, the direct labor resource becomes idle. Furthermore, some revenue is lost due to equipment downtime and time for repair. The overall cost, due to the interaction among the resources within the enterprise, must therefore be evaluated.

By re-engineering to eliminate the conveyor and group the processes at one workstation, the cost can be improved radically without any new investment. In this case, the maintenance crew must be redeployed or the enterprise will not see much economic benefit from process improvements.

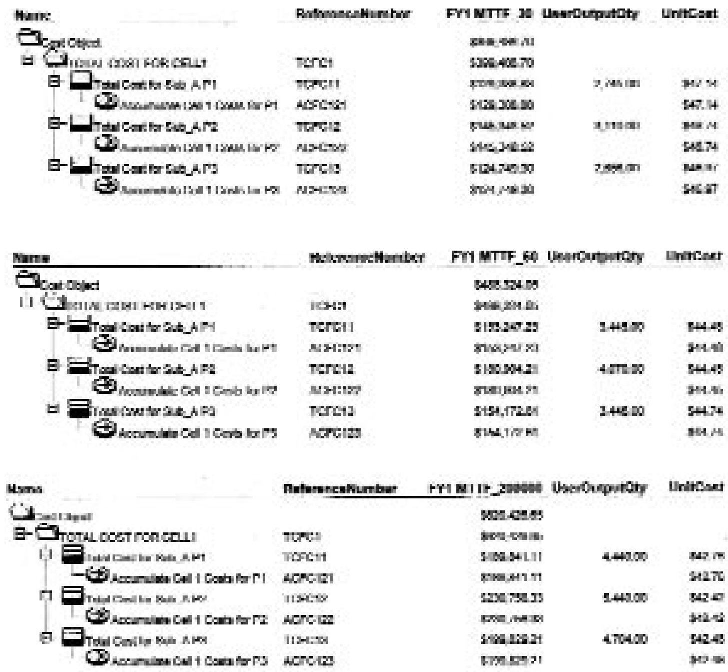


Figure 7. Unit costs for P1, P2, and P3.

The building block for client-server systems

Once the prototype line has been re-engineered, a client-server system (a distributed control system) can be designed. Bar-code scanners connected to a PC can be used to monitor inventory levels of each part for every process at a workstation. The inventory data can be sent over the network to a client's PC at the management office for review. The database, located in a PC server, would have a module for inventory updates. The ABSS must be the heart of the client-server system. It must continuously forecast sources of variability and their economic impact on the business processes to management for remedial action. The actual status on the shop floor (from the scanner data) should be compared with the ABSS data before any action is taken. Furthermore, ABSS should be used for capacity planning of the business processes — for example, the number of workers, the number of machines, and the amount of overtime needed to fulfill a customer order, and the cost associated with the capacity requirements. Again, management should always re-deploy excess labor resources resulting from radically fine-tuning the client-server system. ◆

For further reading

Abbey, Michael and Michael J. Corey, *Orcale 8: A Beginner's Guide*, Oracle Press, 1997.

Buckout, Scott, E. Frey and J. Nemec Jr., "Making ERP Succeed: Turning Fear into Promise," *IEEE Engineering Management Review*, Fall, 1999.

Chandler, Alfred D., Jr., "The Emergence of Managerial Capitalism," *Business History Review*, Volume 58, 1984.

Cokins, G., A. Stratton, and J. Helbling, *An ABC Manager's Primer*, McGraw-Hill, 1992.

Cooper, R., and R. S. Kaplan, *The Design of Cost Management Systems*, Prentice-Hall, 1991.

Hammer, M. and J. Champy, *Reengineering the Corporation*, HarperCollins, 1994.

Hopp, Wallace J. and Mark L. Spearman, *Factory Physics: Foundations of Manufacturing Management*, Richard D. Irwin, 1995.

Kaplan, R. S., and R. Cooper, "How Cost Accounting Distorts Product Costs," *Management Accounting*, April 1988.

Nyamekye, K., and Y. K. An, "Using Rapid Modeling Technology in a Permanent Mold Casting Production Facility," *AFS Transactions*, No. 96-188, 1996.

Kofi Nyamekye, Ph.D., is the owner and president of NRC. He received a B.S. in mechanical engineering from the University of Wisconsin-Madison, an M.S. in mechanical engineering from Pennsylvania State University, and a Ph.D. in industrial and management systems engineering from Pennsylvania State University. As a former educator and a researcher, Nyamekye has published numerous articles and co-authored a book chapter. He has received many awards, among them the 1996 Best Paper Award selected by the Engineering Division of the American Foundrymen's Society. Nyamekye is a certified affiliate of ABC Technologies, a provider of activity-based costing and management information systems, and he has designed a cost model that forecasts the resource and activity drivers for integration with ABCM software. He is a member of the National Institute of Standards and Technology International Consortium of Researchers for Modeling and Simulation Environments for Design, Planning, and Operation of Globally Intelligent Distributed Manufacturing Systems. Nyamekye is a member of IIE, NAMRI/SME, AFS, and ASME. He can be reached by e-mail at kofin@wavecomputers.net.