

INTRODUCTION TO PROCESSMODEL AND PROCESSMODEL 9000

Bruce D. Gladwin

PROMODEL Corporation
1875 South State Street
Suite 3400
Orem, UT 84097, U.S.A.

Charles R. Harrell

Brigham Young University
Manufacturing Engineering
435 CTB
Provo, UT 84602, U.S.A.

ABSTRACT

This paper presents an introduction to ProcessModel[®] simulation software. ProcessModel is a discrete-event simulation tool that combines the power of simulation with the simplicity of flowcharting to aid managers and business analysts who are looking for ways to quantify the effects of variability, uncertainty and resource interdependencies on the performance of complex business processes.

ProcessModel 9000 is a variation of ProcessModel that allows Quality Managers to easily document, analyze and improve the quality processes which are typically reviewed during the ISO 9000 or QS-9000 certification process. ProcessModel 9000 includes ProcessModel software and more than 35 pre-defined flowcharts of the most common ISO/QS-9000 processes.

Both ProcessModel and ProcessModel 9000 aid in the creation of graphical business process models for the purpose of predicting key process performance measures such as cost, throughput, cycle time and resource utilization.

1 INTRODUCTION

The 1993 release of *Reengineering the Corporation* by Michael Hammer and James Champy began a quest at companies throughout the world to examine the ways in which business processes are performed. Rather than simply looking for ways to improve a given process by marginal amounts, the concepts of reengineering require management to challenge the underlying assumptions about their core business processes, with the goal of realizing order-of-magnitude improvements in process performance

The cornerstone of this management revolution is Process Reengineering, an exercise which, in simple terms, means to 1) break a process up into its basic activities, 2) throw out those activities which add no value to the customer, and 3) reassemble the process in a

manner that is most effective and efficient in terms of speed, cost and profitability.

The ensuing proliferation of books and seminars dedicated to process reengineering created a broad demand for software tools that could help document, analyze and improve business processes. Simulation software vendors found themselves in a unique position to provide these tools. Although many existing tools (e.g. manufacturing simulators and simulation languages) were already adept at analyzing production processes, they did so in a manner which was unfamiliar and beyond the technical capabilities of most managers and business analysts. Moreover, detailed cost analysis, one of the critical elements in any process improvement project, was lacking in the majority of off-the-shelf products.

2 PROCESSMODEL OVERVIEW

ProcessModel was developed with the needs of the everyday business analyst in mind. It is first and foremost an excellent process documentation tool. Based on FlowCharter[®], a leading off-the-shelf flowcharting product from Micrografx, Inc., ProcessModel allows flowcharts (and hence process models) to be created in a matter of minutes. Furthermore, the unstructured nature of general purpose flowcharting allows each modeler to build models in a fashion that is most comfortable to him or her.

ProcessModel is also a fully-functional, discrete-event simulation tool based on the robust simulation engine contained in PROMODEL Corporation's high-end products ProModel[®], MedModel[®] and ServiceModel[®]. The common thread that allows the FlowCharter front end to be linked to the ProcessModel simulation engine is OLE Technology. Models are defined graphically in FlowCharter and then translated automatically into a simulation model which can be executed by ProcessModel.

The OLE technology connecting Flowcharter with ProcessModel is bi-directional. That is, communication

between the two applications goes both ways. Through OLE automation, FlowCharter notifies ProcessModel whenever a shape or connection is created, edited or deleted. ProcessModel is thereby able to maintain information on all shapes and connections in a chart. Whenever a shape or connection in the chart is selected, ProcessModel is able to display the appropriate property sheet. Property sheets are used for entering simulation-related information for each of the shapes and connections.

In the other direction, ProcessModel notifies FlowCharter on how to display objects and connections for better visual feedback on the object and connection type. For example, if the user creates a connection between a resource and an activity, ProcessModel knows what types of objects are being connected and instructs FlowCharter to display the connection as a dotted line instead of a solid line. Using OLE technology in this bidirectional fashion, both the chart and the simulation data always remain synchronized.

3 MODELING METHODOLOGY

Models are created graphically by placing shapes in the layout window to represent model objects. ProcessModel makes model editing quick and easy by providing a dialog box (referred to in ProcessModel as a Property Sheet) for each model element. Hence, there is no need to switch back and forth from drawing mode to data mode. Rather, the modeler is free to enter data about each model element as that element is placed on the screen.

The main object types available for building models in ProcessModel include Entities, Activities, Storages, Resources and Routings. Advanced elements such as Attributes, Variables and Action Logic allow for more complex model definition.

3.1 Entities

Entities are the objects which are processed through the system, such as customers, documents, phone calls, or electronic data. Entities are introduced into the process through the Arrivals module which specifies how many and how often entities arrive to the process.

Once an entity has entered the process its future course of action is determined by the processing instructions contained in the activities and routings. Typical instructions include processing delays, move times and entity splits or joins.

Entities that have completed all processing instructions are exited from the system and statistics collected.

3.2 Activities

Inputs are transformed into outputs via activities, which may or may not correspond to physical locations in the system. Activities typically represent processing events and have an associated capacity which limits the number of entities that can be processing at any one time. However, activities can also represent simple decision points, where variables are set, special statistics are collected or routing decisions are made.

Associated with each activity is the logic necessary to describe the actions required to process each entity through each activity in the process. This logic may be as simple as a time delay or as complex as a nested if-then logic statement that chooses one processing time over another based on the attribute values of the entity in process.

The outputs of an activity may or may not consist of the same entity that was input to the process. For example, a batching operation may combine 10 entities of the same or different types into one new entity type.

Activities may also be assigned to shift schedules, thereby limiting the times when an activity may occur.

3.3 Storages

Storages are holding places where entities may wait until requested for further processing downstream. Storages are similar to activities in that they have a capacity. However, whereas the capacity of an activity refers to how many entities may be processing at one time, the capacity of a storage simply refers to how many entities may be holding at that storage at any time.

Storages have the added benefit of allowing the modeler to specify the order in which entities may leave the storage. The options are limited to first-in first-out (FIFO), last-in first-out (LIFO), or no queuing.

3.4 Resources

Resources are agents (typically people or equipment) that are used to process entities at activities and/or to move entities from one activity to another. Activities and routings may require multiple resources, or may choose amongst several available resources based on the occurrence of some user-defined condition.

Like activities, resources may be assigned to shift schedules to limit the availability of the resource.

3.5 Routings

Routings specify how entities travel from one activity to another. Common routing types include: Percentage, Conditional, Alternate, Renege and Attach (for joining

one entity to another). In addition, routings specify the amount of time required for an entity to move from one activity to another and, optionally, the resource required to move the entity.

3.6 Variables

Variables are numeric counters that are used for tracking specific information. Typical applications of variables include collecting user-defined statistics or tracking the number of entities in process from a designated starting point to a designated ending point.

Variables may be either integer or real, and are global in nature. That is, the value of a variable may be checked or set by any entity, at any point in the process.

3.7 Attributes

Attributes are numeric counters which are similar to

variables. However, whereas variables are global in nature, attributes are local. Each entity in the process has a unique attribute value that may be checked or set only by the entity holding that attribute value.

Attributes are typically used for tracking information or for making decisions that are unique to each entity.

3.8 Action Logic

Action logic is used throughout a model to control the events in the process. ProcessModel allows a modeler to

statements and expressions. Action logic is allowed in the arrival, activity, storage and routing property sheets only.

4 EXAMPLE MODEL

Figure 1 shows a typical model that uses all of the modeling elements discussed above. The model represents a software help desk process, similar to the customer support process performed at PROMODEL Corporation in the support of ProcessModel.

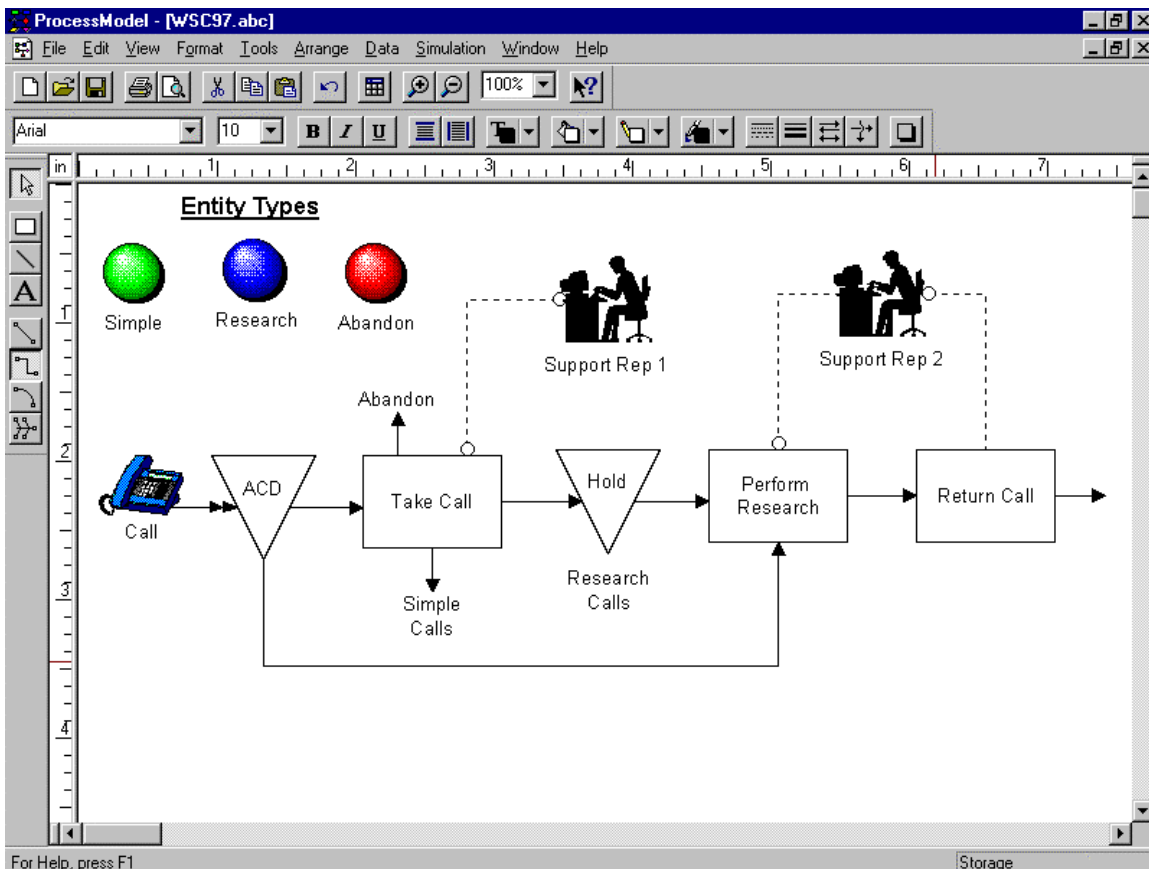


Figure 1: Call Center Model

A model such as this takes only minutes to build in ProcessModel, including the time necessary to enter all of the supporting model data, e.g. activity times, move times, action logic, etc.

4.1 Process Description

The entities in the process include three types of telephone calls: Simple, Research and Abandon. All arriving calls are named simply "Call". Calls enter a storage named ACD where 90% of the arriving calls will be directed to the Take Call activity. The remaining 10% represent callers who are working through an existing problem and know the extension of a particular person in the advanced support group.

The entire process consists of only three activities: Take Call, Perform Research and Return Call. The Take Call activity contains, within itself, an input queue where calls wait until the resource Support Rep 1 is available. Only one unit of Support Rep 1 is available to take initial calls. The time required at the Take Call activity is based upon a triangular distribution with a minimum time of 1.0 minute, a most likely of 3.0 minutes and a maximum time of 6.0 minutes. 80% of the calls entering the Take Call activity are problems of relatively low complexity and thus renamed "Simple". The remaining 20% are calls of moderate to high complexity and are renamed "Research".

Each caller is willing to wait only a certain amount of time for service. If a call cannot be answered before the caller tolerance time expires, then the call will exit the system at the Take Call activity and be renamed "Abandon". The tolerance time is assigned in the arrival action logic according to a triangular distribution with a minimum time of 2.0 minutes, a most likely time of 7.0 minutes and a maximum time of 15.0 minutes. This value is assigned to an attribute which remains with the call throughout the process.

Calls that require research on the part of the advanced support group are sent to a storage named Hold where they wait until a unit of Support Rep 2 is available. There are three units of Support Rep 2. The time to perform research on each call is dependent upon the complexity level of the call. Moderately complex problems will take between 10 and 30 minutes (uniformly distributed), while high complexity problems can take anywhere from 1.0 to 4.0 hours according to a triangular distribution with a most likely time of 1.5 hours.

4.2 Costing

One of the most beneficial features in ProcessModel is the ability to include entity, resource and activity cost factors in the property sheet associated with each

object. This allows for unit costs for each entity type to be calculated automatically as entities exit the process.

Each entity type may have two associated cost factors. The first factor is an initial cost, such as the cost of raw materials or the cost to purchase the entity from an outside supplier. Entities may also have a cost associated with waiting or storage time. Examples include long distance charges for calls placed on hold, or holding costs for inventory in storage.

Resources may also have two associated cost factors: hourly cost and cost per use. These costs are transferred to an entity whenever the entity requires the use of that resource type. For example, if an entity requires the use of a \$12.00 per hour resource for a 1.5 hour activity, then that entity will accumulate \$18.00 each time that activity is performed.

Finally, activities themselves may have cost factors associated with them. The first is a cost per hour of activity time, and the second is a cost per occurrence of the activity. Entities that are processed through an activity will accumulate these costs whenever the activity is performed. These costs are independent of and additional to any resource costs incurred in performing the activity.

As each entity exits the process, its accumulated cost is added to the total cost for that entity type. The total cost is then divided by the number of entities that have been processed to calculate the unit cost.

4.3 Animation and Model Execution

Figure 2 shows a snapshot of the call center model during runtime. Entities arrive along the double headed arrival arrow and travel along routing lines from one activity or storage to another. Both activities and storages have counters below and/or above them to represent the number of entities currently in the activity or storage, or in the input/output queues associated with an activity.

The modeler has a high degree of control over the execution during runtime. For example, the animation speed can be controlled by the slider bar above the animation window, and the clock in the upper right corner can be shown in several formats. Other runtime interaction features include: panning and zooming, event tracing, pausing the execution and suspending the animation for increased execution speed.

One of the most useful features of ProcessModel is the ability to define multiple scenarios for experimentation purposes. The modeler can specify which model parameters to change, along with the desired values, and let ProcessModel automatically run each scenario with as many replications as necessary.

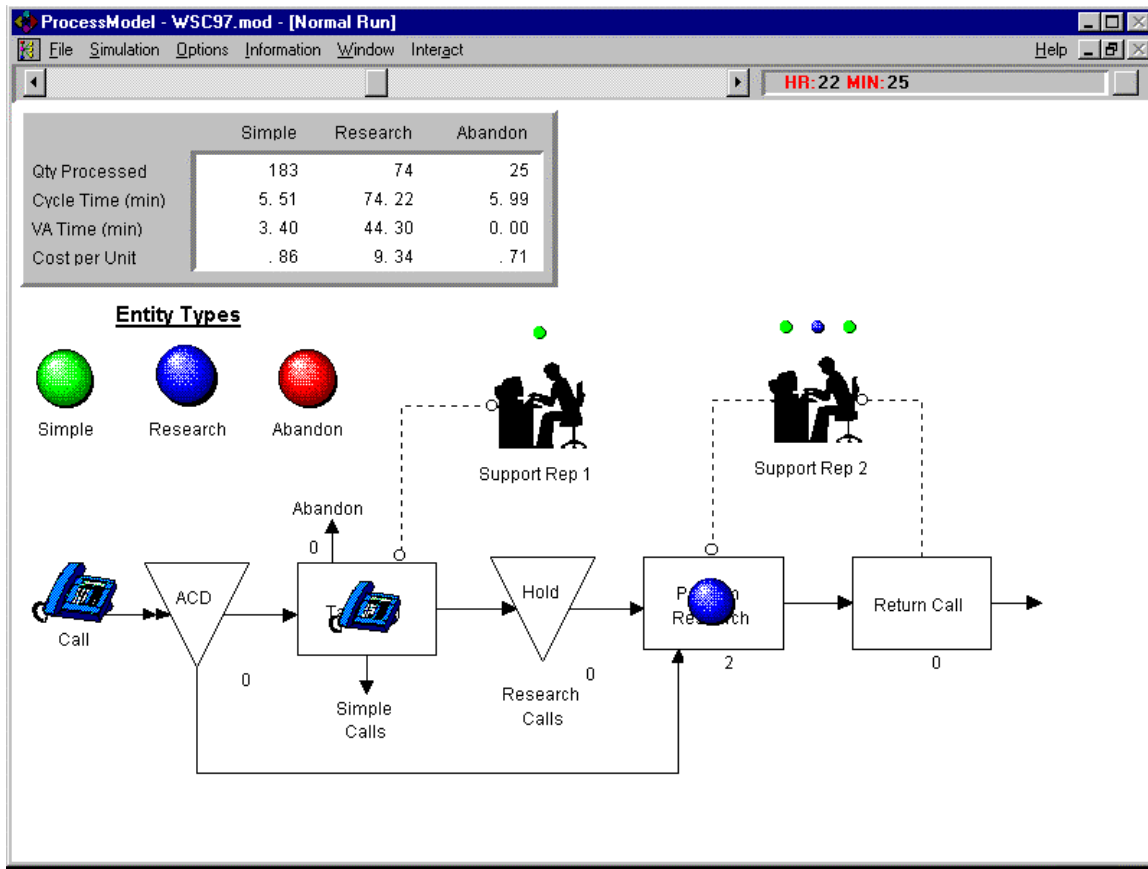


Figure 2: Snapshot of Call Center Model Animation

4.4 Statistics and Output

ProcessModel provides automatic summary statistics in several forms including tables, charts and graphs. If the modeler desires to create his or her own specially formatted reports, raw data can be saved in text files or comma separated files for easy import to other programs such as spreadsheets and databases.

An on-screen scoreboard gives instant feedback on four critical performance measures: Throughput, Cycle Time, Value Added Time and Unit Cost. These measures are updated each time an entity exits the system. Figure 2 shows the scoreboard in the upper left portion of the screen for the example model outlined above. A separate column is created for each entity type for which the modeler chooses to collect statistics.

A detailed, tabular summary is generated at the end of each model run. Figure 3 shows a report for a 40 hour run of the call center example model. The report contains summary statistics for each activity, resource and entity type in the process. Summary statistics are given for the states of each activity, entity type and

resource type. In addition, the percentage of time that each activity, resource type and entity type spent in a particular state is provided.

Charts and graphs, including time series plots of variables, can be created easily in ProcessModel. Bar charts and pie charts can be created for Activity and Resource Utilization, as well as Activity and Resource States. Time series plots, like the one in Figure 4 show how variable values change over time. In this graph the number of calls which abandon over the 40 hour run are compared for two scenarios. The first scenario represents an average call inter-arrival time of 5.0 minutes, whereas the second scenario represents an average inter-arrival time of 4.5 minutes.

Multiple replication analysis is a critical element in understanding the affects of variability on process performance. ProcessModel retains a complete summary report on each replication and automatically generates a multiple replication summary showing the minimum, maximum and average values for all statistics over all runs. Confidence interval calculations are also provided for 90%, 95% and 99% intervals.

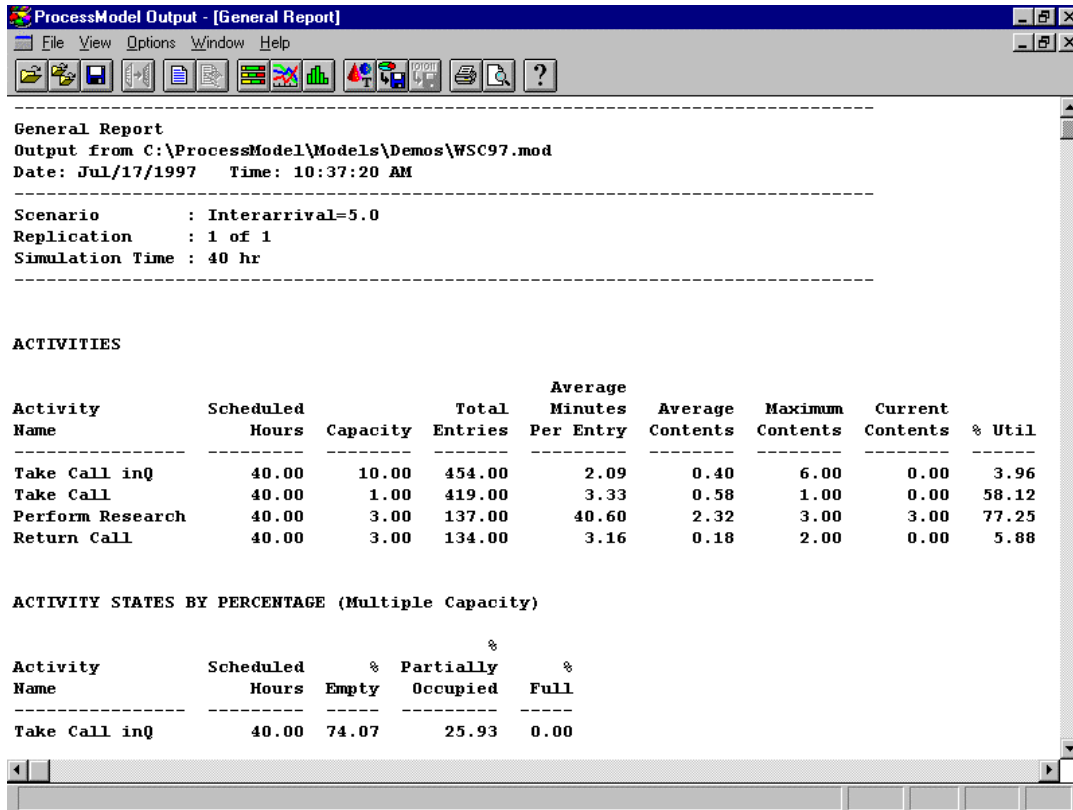


Figure 3: Tabular Summary of Call Center Model Statistics

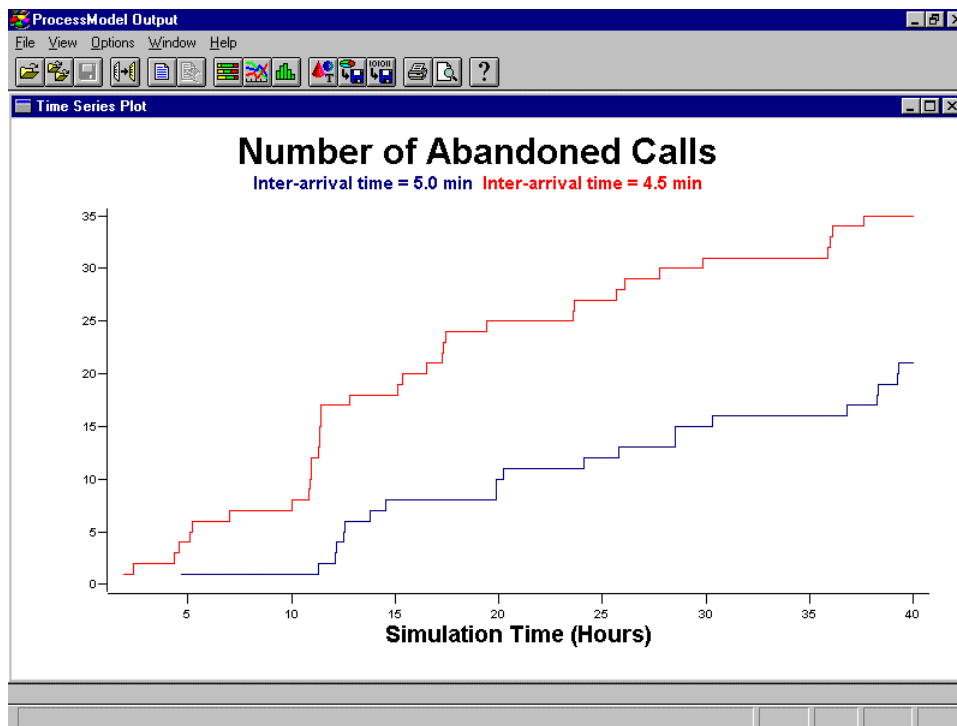


Figure 4: Time Series Plot of Abandoned Calls

5 PROCESSMODEL 9000

Just as the Business Process Reengineering (BPR) revolution has created the need to document, analyze and improve business processes, the ISO/QS-9000 revolution has created the need to document, analyze and improve quality processes. Until recently, this meant opening up a word processor to begin the arduous task of writing down all the steps performed in a quality process. Now, more and more companies are turning to flowcharts to document their quality systems.

According to a recent article in *ISO 9000 News*, surveys show that companies seeking ISO or QS-9000 certification spend only 20% of their time designing and up to 80% of their time writing quality system documentation. Even more alarming is the reality that once these quality manuals are complete, they are seldom used in the day-to-day management of the business due to the sheer volume and impractical nature of text-based documentation.

ProcessModel 9000 is an expansion of the standard ProcessModel product which contains over 35 pre-defined flowcharts of ISO/QS-9000 quality processes. It helps companies pursuing ISO or QS-9000 certification in three ways:

- 1) The simple act of using flowcharts to document quality processes saves time over the traditional method of text-based documentation. It is not uncommon to be able to shrink a 10 page written procedure into a two to three page flowchart.
- 2) The pre-defined flowcharts eliminate the need to "reinvent the wheel" since most companies perform many of the quality processes in a similar nature. The pre-defined charts are simply modified to match a company's specific ways in which they do business.
- 3) The simulation capabilities of ProcessModel allow the processes to be analyzed for inefficiencies and reengineered for increased performance.

6 CONCLUSION

The need for software tools that can be used in the documentation, analysis and redesign stage of process reengineering projects has led to the development of several user-friendly, flowchart-based simulation tools. ProcessModel by PROMODEL Corporation is a tool that links FlowCharter, an off-the-shelf flowcharting product from Micrografx, Inc., with existing high-end simulation technology from PROMODEL.

Variants of ProcessModel are now being developed to meet specific needs in niche markets. The first of these products, ProcessModel 9000, combines the simulation capability of ProcessModel with over 35 pre-defined flowcharts of common ISO/QS-9000 quality processes.

REFERENCES

- Hammer, M., and J. Champy. 1993 *Reengineering the Corporation*. New York: Harper Collins.
- ISO 9000 News*. March, 1997. "Process Mapping Software for ISO 9000".

AUTHOR BIOGRAPHIES

BRUCE D. GLADWIN is Product Manager for ProcessModel software at PROMODEL Corporation in Orem, Utah. He received a B.S. degree in Systems Engineering from the University of Arizona in 1987 and an MBA from Brigham Young University in 1995. Bruce has over 10 years experience in simulation of manufacturing and service systems. His primary responsibilities at PROMODEL include marketing and development of ProcessModel software, and the advancement of simulation modeling in the service industries. Prior to joining PROMODEL, Bruce worked in the advanced manufacturing department at Hughes Aircraft's Missile Systems Division in Tucson, AZ. He is a senior member of IIE and is currently serving as President of the Great Salt Lake Chapter.

CHARLES R. HARRELL is an associate professor of Manufacturing Engineering at Brigham Young University, and founder and chairman of PROMODEL Corporation in Orem, Utah. Dr. Harrell received his B.S. in Manufacturing Engineering Technology from Brigham Young University; M.S. in Industrial Engineering from the University of Utah; and Ph.D. in Manufacturing Engineering from the Technical University of Denmark. Prior to forming PROMODEL, he worked in simulation and systems design for Ford Motor Company and Eaton Kenway Corporation. Dr. Harrell is a senior member of IIE and SME.