Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

PROMODEL TUTORIAL

Charles R. Harrell, Ph.D.
Ken Tumay

Production Modeling Corporation International
1875 South State, Suite 3400
Orem, Utah 84058

ABSTRACT

ProModel combines the flexibility of general purpose simulation languages and the easy-to-use features of manufacturing simulators. This tutorial presents ProModel's design philosophy, describes its modeling elements, and illustrates these elements with an example.

1 DESIGN PHILOSOPHY

In this section, we describe the design criteria used in developing ProModel.

ProModel appeals to a broad user base. ProModel is designed to be used by novice simulation users as well as simulation experts. Industrial and manufacturing engineers have neither the time nor the interest to do programming and yet they have the need for a modeling tool that is powerful enough to simulate a wide range of production systems. ProModel is an easy-to-use and convenient tool for engineers and managers who are interested in quick results. Because of its ease of use, it is also attractive to professors in engineering or business programs who are interested in teaching modeling and analysis concepts rather than teaching programming.

When simulating complex systems that require extensive analysis, usually a simulation expert with programming skills is involved in the modeling activity. In such situations, total modeling flexibility can only be achieved through additional programming. To satisfy this need, ProModel offers complete programming capability in a C or PASCAL type language which can be conveniently accessed without exiting from the program. In this respect, ProModel is powerful and convenient for systems analysts and simulation experts who are interested in ultimate flexibility.

Model development is completely graphical and object-oriented. To the extent possible, all input is provided graphically and information is grouped by objects for quick and intuitive access. This data input approach minimizes the learning curve for beginners and maximizes the efficiency for modifying large and complex models.

Powerful manufacturing constructs minimize model development time. Such typical manufacturing constructs as AGVs, conveyors, cranes, robots are not available in most general purpose languages. On the other hand, those simulators that claim to offer these constructs have rudimentary capabilities that oversimplify the actual hardware characteristics. ProModel offers realistic and flexible constructs for modeling complex manufacturing systems quickly.

Object oriented programming using C++ creates robust and portable software. One of the shortcomings of simulation software has been the difficulty in porting models across a variety of operating systems or hardware platforms. The main reason for these shortcomings is the underlying languages used such as FORTRAN. ProModel takes advantage of the portability and object orientation provided by C++. Furthermore, C++ produces very efficient code that is less prone to bugs and easy to add new features.

The latest advancements in operating systems technology are utilized. ProModel takes advantage of the state of the art memory management techniques, syncronized windowing and dynamic data exchange capabilities offered by WINDOWS and OS/2. Fully compliant with CUA (Common User Access) standards, ProModel allows multiple applications to run concurrently.

Model size is limited only by memory and execution speed is incredibly fast. Model size and execution speed have long been two significant concerns for simulation users on microcomputers. Although those simulation tools running under WINDOWS and OS/2 have eliminated the model size limits, they have done so by sacrificing model size for speed. Unlike other simulation software products, ProModel provides unlimited model size while offering extremely fast execution speed.

Graphics are realistic and easy-to-develop. Realistic looking animation helps simulation to become a powerful communication vehicle between engineers and managers. However, most engineers have neither the time to create 3-D graphics nor the easy access to special graphics terminals. ProModel offers a colorful and powerful 2-D graphics editor with scaling, rotating, etc. capabilities on standard hardware. CAD drawings (e.g. AutoCAD), scanned pictures and drawings can be imported into ProModel for animation development.

Animation development is integrated with model definition. A major drawback of many simulation software products is that animation development is independent from simulation model development. This makes it time consuming and inconvenient for engineers to use animation as a validation\verification tool. ProModel integrates system definition and animation development into one function. While defining routing locations, conveyors, AGV paths, etc., you essentially develop the animation layout. The layout screen is a virtual screen which can be scaled to an actual factory layout.

Simulation results are easy to generate, meaningful and graphical. Most simulation software products require special commands to generate statistics that are difficult to interpret for non-simulationists. ProModel allows quick and convenient selection of reports and provides automatic tabular and graphical reports on all system performance measures. The reports provide meaningful detailed statistics; thus eliminating special commands to generate useful information.

ProModel runs with standard hardware. Most engineers, managers, and professors have easy access to IBM or compatible computers with VGA graphics capabilities. ProModel does not require any special graphics cards, special monitors, or a math coprocessor. This makes it convenient and cost effective for companies and academic institutions that have standard microcomputers. ProModel is also available on LANs.

2 MODELING ELEMENTS

In ProModel, a model defines a production or service system which consists of the items being processed, one or more processing locations, any number of auxiliary resources, routing and operation logic and a production plan or schedule. Modeling elements in ProModel are the constructs used to represent the physical and logical components of the system being modeled.

Physical elements of the system such as parts and resources may be referenced either graphically or by name. Names may be any word or combination of words up to 32000 characters long. Following is a brief description of each of these elements.

2.1 Parts or Entities

Parts or entities refer to the items being processed in the system. These include raw materials, piece parts, assemblies, loads, WIP, finished products, etc. Entities of the same type or of different types may be consolidated into a single entity, separated into two or more additional entities or converted to one or more new entity types.

Entities may be assigned attributes that can be tested in making decisions or for gathering specialized statistics.

2.2 Routing Locations

Routing locations are fixed places in the system (e.g. machines, queues, storage areas, work stations, etc.) to where parts or entities are routed for processing, storage or simply to make some decision about further routing. Route locations may be either single unit locations (e.g. a single machine) or multi-unit locations (e.g. a group of similar machines performing identical operations).

Routing locations may have a capacity greater than one and may have periodic downtimes as a function of clock time (e.g. shift changes), usage time (e.g. tool wear), usage frequency (e.g. change a dispenser after every n cycles), change of material (e.g. machine setup) or based on some user defined condition.

2.3 Resources

A resource may be a person, tooling, vehicle or other device that is used to transport material between routing locations, or to perform an operation on material at a location, or to perform maintenance.

Resources may be any one of the following three types:

General resources which are immediately accessible when not in use and have no special operating characteristics,

Mobile resources which have special operating characteristics to define their movement between tasks,

Robots which have an end of arm tool or gripper,

Cranes (bridge or gantry) which are mounted overhead and move along multiple axes.

2.4 Conveyors

A conveyor is a continuous movement device along which entities are conveyed. Conveyors are accumulating or non accumulating and may be segmented or non-segmented. Conveyors have operating characteristics such as speed and material spacing and may also have downtime characteristics. Conveyors may be configured with transfers, recirculation loops, sortation, accumulation and distribution capabilities. Bi-directional conveyors and complex conveyor networks can be modeled.

2.5 Variables

ProModel provides numerous variables for decision making and statistical reporting. Variables are of two major types: (1) System or state variables such as the clock time or available capacity of a resource, and (2) User-defined variables including gates and multi-dimensional arrays.

2.6 Functions and Distributions

In addition to variables, ProModel has numerous built-in functions and distributions including discrete and continuous empirically defined distributions. Built-in distributions include exponential, normal, uniform, triangular, beta, gamma, erlang, weibull and lognormal.

2.7 Logic and Action Statements

To perform special testing and provide specific instructions within a model, ProModel enables the user to enter logic and action statements. While some statements are related specifically to entities and resources, other statements are general programming statements providing the complete flexibility of a programming language including if-then statements, switch or case statements, loops, complex boolean expressions and file I/O statements. Statement nesting and even subroutines are supported.

3 USING PROMODEL

Much of the ease in using ProModel comes from the simple and straightforward way in which models are defined and results are obtained. A model is defined the same way in which an engineer would naturally describe a production system using the same familiar terminology. A unique "Automatic Model Build" option is available for walking the user through the complete model building process. On-line, context sensitive help screens provide useful hints for data input.

In this section, we will define a problem, step through the model building process, describe the simulation run and present the output results.

3.1 Problem Definition

To illustrate how ProModel works, we define a Flexible Manufacturing Cell with 6 operation stations, a load station, an unload station and an AGV system (see figure 1 for the ProModel layout of the model). *Pallets* are scheduled into an *Input queue* based on an exponential distribution. Each pallet contains 5 *blades*. From the input queue, individual blades are moved to a *Load* station where they are loaded onto a *fixture* by an operator (*Operator 1*). Once the fixture is loaded, an *AGV* is requested to move it to *Machine A* or *Machine B (Station 1)*. From either Machine A or B, a fixture is sent to either *Machine C* or *Machine D (Station2)*. After processing on Machine C or D, the fixture is sent to an on-line *Inspection* station. If the blade passes inspection, the fixture is sent to the *Unload* station. If the blade fails, it will be sent to a *Rework* station, repaired and then sent to Unload. The probability of having a reject is .045. At the Unload station, the blade is taken off the fixture by an operator (*Operator 2*). The empty fixture is sent to the *Fixture queue* waiting to be loaded with another blade. The system is modeled with 5 fixtures and 3 AGVs.

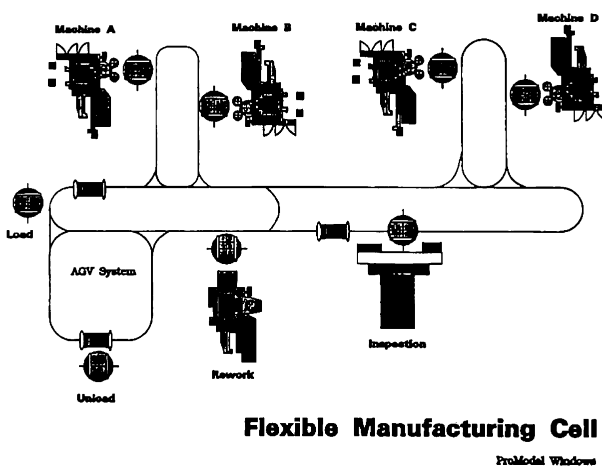

Figure 1: A ProModel Layout Screen

3.2 Building The Model

To build the model, first the routing locations such as the load stations and machines are defined and placed on the layout. Operations performed for each part type at each location is then entered. After the routing locations and operations have been defined, auxiliary resources including the operators and AGV 's are defined. With all of the locations and resources identified, the parts or entities and their associated routing through the system can be defined. Finally, the scheduling of parts into the system and simulation parameters are defined.

3.3 Defining The Model Locations

Locations are defined by choosing the *Location* module from the *Model Definition* menu and entering a name, capacity and number of units for each of the locations in the system. Operations performed at each location are then defined.

To place a location on the screen layout, a library or user drawn graphic is selected (see Figure 2) and dropped on the screen layout. These graphics may be moved, sized and even rotated. Any number of additional graphics (e.g. labels, counters, etc.) may be defined for the location.

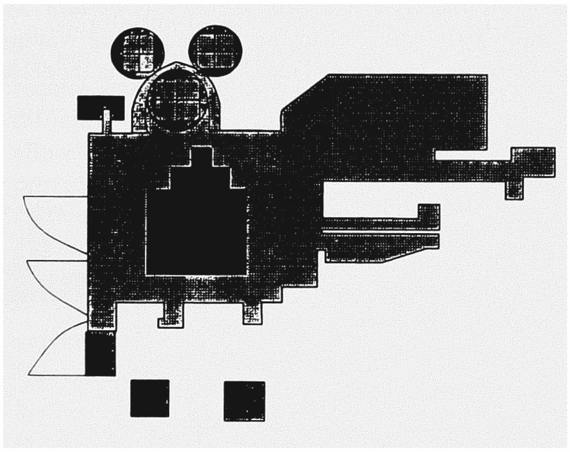

Figure 2: A Detailed View of An Icon

This particular model has the following location definitions:

Location	Capacity	Units
Fixt Que	10	1
Input Que	20	1
Load	1	1
Station A	1	2
Station B	1	2
Inspect	1	1
Rework	1	1
Repair	1	1
Unload	1	1

Operations at a location are defined based on part type. The following operations are defined for this model:

Part	Location	Operation or Action
fixture	Fixt Que	0
pallet	Input Que	Split 5 blade
fixture	Load	Join 1 blade Use Op 1 for 1
fixture	Station A	2
fixture	Station B	3
fixture	Station C	1
fixture	Rework	U(3, 1.5)
fixture	Unload	Use Op 2 for U(1, .5)
fixture	Inspect	N(1.5, .25)

3.4 Defining Auxiliary Resources

Additional resources required for processing or material handling are defined through choosing the *Resource* module and entering the name and any operating characteristics for each resource.

The operators in this model are considered to be general resources that, at least for modeling purposes, are assumed to be immediately accessible when available. For general resources, only the number of units and screen position need be defined. This model uses two general resources (Operator 1 and Operator 2) consisting of 1 unit each.

The AGV is a mobile resource which requires that a path network be defined with speeds and distances. Location interface positions and other control points are defined as well as any work search priorities. Idle vehicle deployment may also be specified. The AGV has a path layout that is defined graphically with distances either automatically supplied according to scale or manually entered. The operation parameters for the AGV are as follows:

Speed ft(m)	Pickup (sec)	Deposit (sec)	Search rule	Acc.,Dec (fpss)
120	4	4	OLDEST	3

3.5 Defining Parts and Associated Routings

Parts are defined by invoking the *Part* or *Entity* module and entering a name for each part type. A predefined or user-defined graphic may also be selected to represent the part. For each part type, a routing is defined by selecting a "from" location and one or more alternative "to" locations. Locations are selected either by entering the location name, selecting the location from a help menu or by simply clicking the mouse on the location. Any location selection rules are entered as well as any move time or movement resource used to make the move.

For our demonstration model, there are three part types defined: fixture, pallet and blade.

The routing for *fixture* is defined as follows:

From	To	Rule	Resource	Time
Fixt Que	Load	0	0	1.5
Load	Station A	0	AGV	0
Station A	Station B	0	AGV	0
Station B	Inspect	0	AGV	0
Inspect	Rework	95.5%	0	.5
Inspect	Unload	4.5%	AGV	0
Rework	Unload	0	AGV	0
Unload	Fixt Que	0	0	.5

Note that no time is specified for movement of the AGV since it is a mobile resource whose time is determined by the speeds and distances defined in the Path Logic module.

The routing for *blade* is as follows:

From	To	Rule	Resource	Time
Input Que	Load	JOIN	0	:30

Note that there is no routing specified for *pallet* since it changes name after splitting.

3.6 Scheduling Part Arrivals

With all of the locations, resources and part routings defined, the only thing left is to give the the system work to perform. This is done by defining the part arrivals in the *Arrivals* module. Each part arrival is defined by entering a part name or selecting a part from the parts menu. The location where the parts are to arrive is then selected by name entry or clicking on the location. The quantity per arrival and frequency of arrivals are also defined.

3.7 Running the Simulation

The length of the run and other simulation options are defined in the *General Information* module. The simulation can be run with or without animation. The animation may even be only temporarily disabled to speed ahead. Additionally, the state of the system can be saved at any point in time which can be recreated instantly without having to run the entire simulation again.

3.8 Analyzing the Results

The results of the simulation run contain resource utilization, queueing, and throughput statistics. More detailed statistics for producing histograms and time series charts can be collected.

4 CONCLUSIONS

Until recently, manufacturing companies have not fully benefited from simulation in making continuous improvements because of the time, programming expertise, and cost involved in getting useful results. ProModel is designed for manufacturing companies to fully achive the benefits of simulation technology at an affordable price. ProModel is directed toward making simulation a standard tool in the hands of engineers, managers and systems analysts just as spreadsheet software is in the hands of accountants and financial analysts.

REFERENCES

Harrell, C. R.. 1989. PROMOD (PROduction MODeler) for IBM PC's. In *Supplementary Proceedings of the SCS Multiconference,* eds. S. Spencer and G.Richardson, 65-70. San Diego, California.

Harrell C. R. and K. Tumay. 1990. ProModelPC Tutorial. In *Proceedings of the Winter Simulation Conference,* eds. O. Balci, R. Sadowski and R. Nance, 128-131. New Orleans, Louisiana.

Harrell C. R. 1990. Trends in Manufacturing Simulation. In *Proceeding of Autofact Conference,* eds. A. Adlard, 21-31. Detroit, Michigan.

Tumay, K. 1989. Opportunities and Challenges for Busy Engineers In Manufacturing Simulation (Panel). In *Proceedings of theWinter Simulation Conference,* eds. E. MacNair, K. Musselman and P. Heidelberger, 859-864 Washington D.C.

Production Modeling Corporation of Utah. 1991. *ProModelPC User's Manual.*

AUTHOR BIOGRAPHIES

CHARLES R. HARRELL, Ph.D., is the founder and President of Production Modeling Corporation International. Charles has received his B.S. in Manufacturing Engineering Technology from Brigham Young University, M.S. in Industrial Engineering from University of Utah and Ph.D. in Manufacturing Engineering from the Technical University of Denmark.

KEN TUMAY received his B.S. and M.S. degrees in Industrial Engineering from Arizona State University. Prior to joining PMCI, Ken worked for The Confacs Group and CACI Products Company. He is a member of IIE and SME