

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Industrial and Management Systems
Engineering Faculty Publications

Industrial and Management Systems
Engineering

6-1992

Introduction to IntelliSIM 1.0

Paul Savory

University of Nebraska at Lincoln, psavory2@gmail.com

Follow this and additional works at: <https://digitalcommons.unl.edu/imsefacpub>



Part of the [Industrial Engineering Commons](#), [Operational Research Commons](#), and the [Other Operations Research, Systems Engineering and Industrial Engineering Commons](#)

Savory, Paul, "Introduction to IntelliSIM 1.0" (1992). *Industrial and Management Systems Engineering Faculty Publications*. 70.

<https://digitalcommons.unl.edu/imsefacpub/70>

This Article is brought to you for free and open access by the Industrial and Management Systems Engineering at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Industrial and Management Systems Engineering Faculty Publications by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

IntelliSIM

NIST Project Notebook

June 1992

Systems Simulation Laboratory
College of Engineering and Applied Sciences
Arizona State University
Tempe, AZ 85287-5906

IntelliSIM DEVELOPMENT TEAM

The following individuals have participated in the development of the software and documentation for IntelliSIM 1.0.

Faculty:

Dr. Gerald Mackulak

Dr. Jeffery Cochran

Graduate Research Associates:

Enrique Del Castillo

Ju-Hyun Kim

Chung-Horng Lung

Paul Savory

Graduate Research Assistants:

Jin-Yng Hwang

Sanjay Mehta

Qi Wan

Umesh Manathkar

COPYRIGHT

Copyright, 1992, by the Systems Simulation Laboratory and Arizona State University.

All right reserved worldwide. No part of this publication or included software may be reproduced, transmitted, transcribed, or stored in a retrieval system or translated into any human or computer language in any form or by any means electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the expressed written permission of the Systems Simulation Laboratory.

DISCLAIMER

IntelliSIM is distributed on an “**AS IS**” basis without warranty.

The Systems Simulation Laboratory makes no warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no event will the Systems Simulation Laboratory be liable for consequential damages even if the Systems Simulation Laboratory has been advised of the possibility of such damages.

ORGANIZATION OF THIS MANUAL

This user's manual and reference guide provides both an introduction to IntelliSIM and a detailed discussion on its features. The guide is divided into seven chapters:

- Introduction
- Research Concepts In IntelliSIM
- Installation Procedure
- Getting started with IntelliSIM
- Model Selection
- Model Configuration
- Execute Simulation

As part of this preface section, we have included a glossary of terms which explains the terminology used in the IntelliSIM environment. In addition, a list of all the figures and tables included throughout the guide is also provided for easy and quick reference.

The first four chapters provide an introduction to IntelliSIM and the research concepts it is based on. We recommend that all users read them before proceeding in using IntelliSIM. Chapter 4 provides a good introduction to IntelliSIM and shows how to quickly start using the environment. Chapters 5 and 6 explain how to create and configure models in IntelliSIM. These are detailed chapters on how to use features in IntelliSIM. Once a model is configured, Chapter 7 describes how to run and analyze simulation experiments using IntelliSIM.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	ix
LIST OF TABLES	xii
LIST OF APPENDICES	xiv
GLOSSARY OF TERMS AND PHRASES	xvi
 1. INTRODUCTION	 1
1.1. What is IntelliSIM?	1
1.2. Systems Simulation Laboratory	1
1.3. History of IntelliSIM	2
1.4. Systems Simulation Laboratory Publications	5
 2. RESEARCH CONCEPTS IN INTELLISIM	 11
2.1. Introduction	11
2.1.1. Expert System Simulation.....	11
2.1.2. Model Creation	12
2.1.3. A Microcomputer Based Approach	13
2.1.4. Planning and Design Issues	14
2.2. Discrete-Event Simulation	16
2.2.1. Steps in a Simulation Study	16
2.2.2. Classification of Simulation Tools	18
2.3. The "Perfect" Simulation Environment	21
2.4. Critical Configuration Issues	24
2.4.1. The Topology of Expert Simulation	24
2.4.2. Taxonomical approaches to Simulation	25
2.5. Automation of Simulation Model Development	26
2.5.1. The GT Scheme	26
2.5.1.1. GT Scheme Form	26
2.5.1.2. Generic Models Structure	27
2.5.1.3. Naming Generic Model Files	28
2.5.2. Automated Analysis of Univariate Statistics	33
2.5.2.1. Traditional Input Data Analysis	33
2.5.2.2. Parameterization of Distributions	34
2.5.2.3. Automated Analysis Methodology	35
2.6. Fundamental Concepts of Generic/ Specific Modeling	39
2.7. IntelliSIM Architecture	41

2.7.1. Methodological Approach	41
2.7.2. Architecture Specification	43
2.7.2.1. Design Goal Formulation	45
2.7.2.2. Generic Model Selection	46
2.7.2.3. Specific Model Generation	46
2.7.2.4. Execution of Experiments	46
2.7.2.5. Output Interpretation and Validation	47
2.7.2.6. Documentation and Implementation	47
2.8. References	48
3. INSTALLATION PROCEDURE	51
3.1. Distribution Disks	51
3.2. Hardware Requirements	51
3.3. Software Requirements	52
3.4. SIMAN Requirement	52
3.5. Library File Requirement	52
3.6. Installing IntelliSIM	53
4. GETTING STARTED WITH IntelliSIM	55
4.1. Executing IntelliSIM	59
4.2. Using the Menus	59
4.3. Saving Specific Models	60
4.4. Helpful Hints	60
4.5. Tutorial/Example of IntelliSIM Session	62
5. MODEL SELECTION	77
5.1. Generic Model Selection by Name	80
5.2. Specific Model Selection	83
5.3. Generic Model Selection by Consulting the Expert System	85
5.3.1. Meta-Level Query	85
5.3.2. Assumption Review and Measuring the User Acceptance	89
5.3.3. GT Code Generation and Model Selection	89
6. MODEL CONFIGURATION.....	95
6.1. Configuration	95
6.2. Work Stations	97
6.2.1. Addition of a New Work Station	97

6.2.2. Update of Existing Work Stations	99
6.3. Work-In-Process Storage Areas	105
6.4. Optional and Special Resources	112
6.5. Material Handling System	118
6.6. Jobs	125
6.6.1. Addition of a New Job Type	125
6.6.2. Update of Existing Job Types	127
6.7. Operation Routings	141
6.8. Equipment Breakdown/Maintenance	148
6.9. Fit Data Distribution	154
6.10. Output Statistics	159
6.11. Model Status Summary	163
6.12. Exit Model Configuration	163
7. EXECUTE SIMULATION	165
7.1. Manual Execution of Specific Model	167
7.2. Semi-Automatic Experimental Design	171
7.3. Viewing and Editing the Summary Report	176

FIGURES

2.1	Steps in a Simulation Study.....	16
2.2	An Extension to Schmidt's Taxonomy of Simulation Tools.....	19
2.3	Flow of Jobs Through a Generic Model	32
2.4	Automated Distribution Fitting	36
2.5	Plot of β_1 verses β_2	37
2.6	Specification of an Architecture for Intelligent Simulation Environments	44
4.1	Menu Structure if IntelliSIM	55
4.2	Physical Layout of the Manufacturing Cell	64
4.3	Main Menu.....	66
4.4	Select a Model	66
4.5	Configure Selected Model	68
4.6	Exit Model Configuration	72
4.7	Execute Simulation	73
4.8	Select Semi-Automatic Analysis Option	74
4.9	Confidence Intervals and Percentage Error for Each Batch	75
4.10	Graphic Summary of Semi-Automatic Analysis	75
4.11	Summary Report	76
5.1	First Introductory Screen of IntelliSIM.....	78
5.2	Main Menu.....	78
5.3	Model Selection Menu.....	79
5.4	Generic Model Names Menu	81
5.5	Description of a Generic Model	81
5.6	Job Arrival Patterns Menu for Generic Model Selection by Name	82
5.7	IntelliSIM Prompt for Specific Models Disk	84
5.8	Specific Models Menu	84
5.9	Introductory Screen for IntelliSIM Expert System	86
5.10	Introductory Screen for Meta-Level Query.....	86
5.11	Meta-Level Query Questions	87
5.12	Meta-Level Query Explanation Mechanism.....	87
5.13	Successful Conclusion of Meta-Level Query.....	88
5.14	Unsuccessful Conclusion of Meta-Level Query.....	88
5.15	Introductory Screen for Assumption Review	90
5.16	Assumption Review Menu	90
5.17	Sample Screen for Assumption Review Menu	91
5.18	Introductory Screen for Model Selection Phase	91
5.19	First Model Selection Menu.....	92
5.20	Second Model Selection Menu.....	92
5.21	Conclusion of the Model Selection Phase.....	93
6.1	Model Status Summary Screen	96
6.2	Model Configuration Menu.....	96
6.3	N/A - No Menu or Figure	
6.4	Work Stations Menu.....	98

6.5	Addition of a New Work Station	98
6.6	Work Stations Update Menu.....	100
6.7	Changing the Station Types.....	100
6.8	Changing the Primary Resource Capacities.....	102
6.9	Changing the First Optional Resource Assignments.....	102
6.10	Changing the Second Optional Resource Assignments	103
6.11	Processing Time Distributions Menu	103
6.12	Changing the Processing Time Distributions.....	104
6.13	Changing the Efficiency Ratios.....	106
6.14	Changing the Alternatives for Machining Stations	106
6.15	Changing the Failure Rates for Inspection Stations	107
6.16	WIP Storage Areas Menu.....	107
6.17	Changing the Input Buffer Capacities	108
6.18	Changing the Output Buffer Capacities	110
6.19	Attempting to Change the Output Buffer Capacities When They Do Not Exist.....	110
6.20	Balking Options Menu.....	111
6.21	Priority Rules Menu.....	111
6.22	Optional and Special Resources Menu	113
6.23	Attempting to Change the Optional Resource Capacities When They Do Not Exist	113
6.24	Changing the First Optional Resource Capacities	114
6.25	Changing the Second Optional Resource Capacities	114
6.26	Changing the Pallet Capacities.....	116
6.27	Changing the Number of Parts Per Pallet Type	116
6.28	Changing the Fixture Capacities	117
6.29	Changing the Number of Parts Per Fixture Type	117
6.30	Material Handling System Menu	119
6.31	Changing the Manual Transportation Distribution.....	119
6.32	Transporters Menu and Changing the Features of Transporters.....	120
6.33	Entering the Starting Station for Transporter Distances	120
6.34	Changing the Transporter Distances	121
6.35	Conveyors Menu and Changing the Features of Conveyors	123
6.36	Entering the Starting Station for Conveyor Segments	123
6.37	Changing the Conveyor Segments	124
6.38	Jobs Menu.....	126
6.39	Jobs Update Menu.....	128
6.40	Specification of Number of Batches Per Job Type for Arrival Pattern A0.....	128
6.41	Job Arrivals Menu for Arrival Pattern A1	129
6.42	Choice of Job Arrival Distribution for Arrival Pattern A1	129
6.43	Specification of Cumulative Percentages of Job Types for Arrival Pattern A1	130
6.44	Job Arrivals Menu for Arrival Pattern A2	132
6.45	Specification of Job Arrival Distribution and Parameters for Arrival Pattern A2.....	132
6.46	Job arrivals menu for Arrival Pattern A3	133
6.47	Specification of Number of Batches Per Job Type for Arrival Pattern A3	133
6.48	Specification of Arrival Times of Batches for Arrival Pattern A3.....	134

6.49	Changing the Pallet Types	136
6.50	Changing the Fixture Types	136
6.51	Changing the Batch Sizes	137
6.52	Changing the User assigned Priorities.....	137
6.53	Changing the Due Dates for Arrival Patterns A0 and A3.....	139
6.54	Attempting to Change the Due Dates for Arrival Patterns A1 and A2.....	139
6.55	Changing the Match Numbers for Arrival Patterns A0 and A3.....	140
6.56	Changing the Match Numbers for Arrival Patterns A1 and A2.....	140
6.57	Operations Routing Menu.....	143
6.58	Defining a New Operation Routing for a Job.....	143
6.59	Specifying Processing Time Parameters for a New Operation Routing.....	144
6.60	Adding a New Operation to an Existing Routing	146
6.61	Deleting an Operation from an Existing Routing	146
6.62	Change Operations Menu.....	147
6.63	Changing Process Time Parameters for an Existing Operation	147
6.64	Changing Material Handling Equipment Type.....	149
6.65	Equipment Breakdown/Maintenance Menu	149
6.66	Changing Time Between Failures Distribution	151
6.67	Specification of Equipment Breakdowns	151
6.68	Specification of Time Between Failures Distribution Parameters	152
6.69	Specification of Scheduled Maintenance	152
6.70	Specification of Time Between Maintenance Distribution Parameters	153
6.71	Fit Data Distribution Menu	155
6.72	Data Fitting Editor	155
6.73	Data File Name Prompt.....	156
6.74	General Statistics of Data Set	156
6.75	Feasibility Rules	157
6.76	Data Fitting Conclusion.....	157
6.77	Data Fitting Summary Report.....	158
6.78	Output Statistics Menu	160
6.79	Specifying Statistics for Work Stations	160
6.80	Specifying Statistics for Optional Resources	162
7.1	Execute Simulation Model	166
7.2	IntelliSIM Prompt for Specific Models Disk to Select a Model for Manual Execution	168
7.3	Specific Models Menu to Select a Model for Manual Execution	168
7.4	Simulation Run Length and Number of Runs for a Specific Model	169
7.5	Sample SIMAN Summary Report Generated by Manual Execution of a Specific Model	170
7.6	Selection of Type of Priority Variable	172
7.7	Selection of the Name of Priority variable from the List.....	172
7.8	Table Showing the Confidence Interval and Percent Error for the Priority Variable in Real Time by Semi-Automatic Analysis.....	173
7.9	Graphical Display of Results at the End of Semi-Automatic Analysis	174
7.10	Sample Summary Report generated in Semi-Automatic Analysis.....	175

TABLES

2.1	The Perfect Simulation Environment.....	22
2.2	Generic Modules	30
2.3	Comparison of New Approach and Old Approach to Discrete Simulation of Manufacturing Processes	42
2.4	Contents of Knowledge Base and Data Base in Intelligent Simulation Environment Architecture	45
4.1	IntelliSIM Menu Options.....	56
4.2	Processing, Loading and Unloading Time for Each of the Stations in the Manufacturing Cell	65

APPENDICES

A	Contents of (ISIM) IntelliSIM Diskettes	177
---	---	-----

GLOSSARY OF TERMS AND PHRASES

- **Batch** - a grouping or consolidation of parts. There may be multiple batches of the same job type flowing through the system at any time.
- **Batch size independent process** - a type of process which is independent of batch size, where multiple jobs can be processed all at once. For instance, heat treatment processes, processes where multiple jobs are treated simultaneously in the same oven, or spray paint processes.
- **Deterministic interarrival times** - the time between a jobs arrival is fixed or is constant (it has no variation).
- **Efficiency ratios** - a measure of the efficiency of a workstation. If the efficiency ratio is less than 1, service time will be longer than the standard.
- **Experimental frame** - defines the experimental conditions under which the model is run to generate specific output data. It specifies the set of parameters and input data to execute the model.
- **.EXP files** - files with the .EXP extension store the experimental frame (experimental conditions) of a model.
- **.FAC files** - files with the .FAC extension contain the model structure.
- **First optional resource** - two types of optional resources per station are allowed in generic models in IntelliSIM. These are simply referred as the first optional resource and the second optional resource.
- **Fixture** - fixtures are allocated to jobs at a loading/unloading station. A job remains with a fixture during its routing and is released when the job is routed back to a loading/unloading station (for unloading).
- **Generic model** - a model that has been designed to meet the needs of several different types of manufacturing situations. It is a prior model flexible enough to fit a wide range of applications. Generic models are populated with data to convert them into specific models.
- **Job** - a job is a batch of parts visiting a number of work stations for various processing steps. Job types are distinguished by their operation routing.
- **Match numbers for assembly** - a unique product identification. For instance, two parts with the same match number can be matched/combined together.
- **Operation routing** - sequence of stations that a job visits for processing.

- **Pallet** - used for transporter purposes only. A pallet is allocated to a job after the processing of that job is finished and it remains seized during transportation of the job to the next station, where it is released.
- **Primary resources** - the identical servers associated with a work station. A job visiting a work station has to seize one of these servers to start processing.
- **Push/Pull systems** - two different types of manufacturing systems. In a "Push" system, the facility produces to meet some demand forecast. In a "Pull" system the facility produces according to actual orders. This "triggers" the production process downstream.
- **Random interarrival times** - the amount of time between two jobs of the same type arriving or to be processed in some workstation is random. It is usually described with a probability distribution.
- **Random number streams** - controls the random numbers used in a simulation model. Randomness (or chance) is simulated in a computer by a sequence of numbers which are not known to the user and in this way mimic random behavior.
- **Optional resources** - an resource that may be needed for processing a job at some work station. Optional resources can be dedicated to a station or shared by multiple stations. Examples of optional resources are: human operators, robots, and tools such as jigs and cutters.
- **Second optional resource** - two optional resources per station are allowed in generic models in IntelliSIM. These are simply referred as the first optional resource and the second optional resources.
- **Sequence-dependent setup times** - a manufacturing situation in which a machine that produces a variety of parts takes different amounts of time for setting up each type of part (depending on the order in which the parts are processed). When the times are the same disregarding order, the setup times are sequence independent.
- **Segments** - a segment is a section of a conveyor path that connects two work stations.
- **Specific model** - a model that has been configured to represent a particular manufacturing system by adding details about the system such as arrival rates, job types, routings, etc.
- **Steady-state simulation** - A non-terminating simulation. A simulation which has no event specifying the end of the simulation. In this type of simulation, the long-run behavior of the system is analyzed, without consideration of the initial conditions.
- **Terminating Simulation** - a simulation which has a fixed starting condition and some type of event defining the natural end of the simulation. For example, such an event could be the close of the business day.

- **Time persistent statistic (time dependent variable)** - a variable of interest which is a function of the simulated time for, example, the average work in process inventory in front of a work station.
- **Transporters** - material handling equipment, such as AGVS, carts, cranes, etc.
- **WIP** - work in process inventory.
- **Work station** - any resource that may be found in a discrete manufacturing system which is not a material handling resource. A work station is a basic shop floor unit consisting of a number of identical parallel servers. There are three types of stations: machining, assembly and inspection.

CHAPTER 1

INTRODUCTION

1.1. What is IntelliSIM?

IntelliSIM is a prototype for a new generation of knowledge-based simulation tool that has been developed by the Systems Simulation Laboratory at Arizona State University. This tool is a computer environment that allows non-simulation trained modelers to predict the performance of a manufacturing system for which the necessary data is available. The system provides predictive data on such items as throughput time, queue levels, equipment utilization, reactions to machine failures, etc.

With IntelliSIM, the benefits of discrete-event simulation can be exploited without requiring the high level of expertise necessary to successfully conduct a sound simulation study. The approach offered with IntelliSIM is one which will offer substantial time and savings over currently available simulation tools.

1.2. Systems Simulation Laboratory

The Systems Simulation Laboratory (SSL) at Arizona State University was formed in 1987 by Drs. Jeffery K. Cochran and Gerald T. Mackulak. They created the laboratory to concentrate efforts and resources in the College of Engineering and Applied Sciences relating to simulation analysis. The goal of the SSL is to establish leadership in the development of simulation techniques and in the conceptualization of the next generation of simulation software tools to solve real problems.

A student simulation affiliate program operates in conjunction with the SSL. This affiliation program was created to provide a structure in which students with a common interest in simulation may pursue research topics in a structured manner. The program provides a forum by which the students can compare ideas on research topics which apply

to the laboratory and to their research objectives. Students in the program are currently divided equally between Master's and Doctoral candidates.

The reader is referred to Section 1.5, "System Simulation Laboratory Publications" for a listing of recent research publications by the Systems Simulation Laboratory.

1.3 History of IntelliSIM

MASCOT: An Initial Implementation

MASCOT - Motorola Artificial Simulation Companion Organizer and Teacher

Project Sponsor: Motorola Manufacturing Management Institute

Project Title: Generic Simulation on the PC Project

Project Years: 1985 - 1987

Principal Investigators: G.T. Mackulak, J.K. Cochran, and D.L. Shunk
Systems Simulation Laboratory

Funding Level: \$178,790

Graduate Assistants Doctoral Candidates: Li Lin
Master's Candidates: Dave Castillo, Eric Du, Chin-Yuh Lin,
Ying-Che Yang

Initial concepts on generic models and intelligent simulation environments are included in MASCOT (**M**otorola **A**rtificial **S**imulation **C**ompanion **O**rganizer and **T**eacher) developed for Motorola, Inc. This system is a simulation environment capable of running simulations of discrete semiconductor manufacturing factories. Currently, MASCOT is in use throughout Motorola for simulation training.

Two basic concepts are involved in MASCOT: generic-specific models and expert system design control. The concept of necessary expert assistance has been clearly demonstrated by the prolific growth of modelling consulting firms. Most simulation practitioners are in need of expert assistance when faced with a modelling effort. MASCOT was aimed at automating this process of expert assistance for the specific

application of electronics' manufacture (fabrication and assembly are two.) This package concentrates on three fundamental problems common to simulation methodology: simulation training, generic model development and application of expert knowledge. The particular application of our system (electronics manufacturing), however, helps specify some structure to what are otherwise nebulous issues.

GUIDES: Refinement of Research Concepts

GUIDES - Generic User-Friendly Intelligent Design Environment for Simulation

Project Sponsor: National Science Foundation
Institute for Manufacturing and Automation Research

Project Title: Intelligent Simulation Systems for Automating Factories

Project Years: 1987-1991

Principal Investigators: J.K. Cochran and G.T. Mackulak
Systems Simulation Laboratory

Funding Level: \$298,000

Graduate Assistants: Doctoral Candidates: Suck-Chul Hong, Li Lin, Nur Ozdemirel,
Edward Stahlman
Master's Candidates: Lotfi Gaafar, Jordan Snyder

Concepts implemented in MASCOT have been further expanded in the development of GUIDES (**G**eneric **U**ser-friendly **I**ntelligent **D**esign **E**nvironment for **S**imulation).

Among the current trends in simulation, most of the existing systems have difficulties in modeling the system and in providing an intelligent system environment. The primary problem is that modelers view all simulation models as unique, and tend to recreate topologically similar systems when only the parameters are different. Another problem in simulation environments is that the current systems do not provide user design goals. The requirement in providing user design goals is important, for experimental design selection and specific model structure are affected by these goals.

To develop GUIDES, simulation theory in both the computational and knowledge domains have been extended. Specific advances include: simulation design goal formulations, shop floor decision support features, semi-automatic statistical input and output analysis, and analytical metamodels of production line transient behavior.

IntelliSIM: Commercialization of Research Concepts

IntelliSIM - Intelligent Simulation

Project Sponsor: National Institute of Standards and Technology
Institute for Manufacturing and Automation Research

Project Title: Intelligent Simulation Environment Testing and Commercialization Plan

Project Years: 1992 - Current

Principal Investigators: G.T. Mackulak and J.K. Cochran
Systems Simulation Laboratory

Funding Level: \$175,000

Graduate Assistants: Doctoral Candidates: Enrique Del Castillo, Ju-Hyun Kim,
Chung-Horng Lung, Paul Savory

Master's Candidates: Jin-Yng Hwang, Sanjay Mehta, Qi Wan
Umesh Manathkar

The latest implementation of research concepts has resulted in IntelliSIM (**Intelligent Simulation**). Development of IntelliSIM is focused on determining what technology and features are required in an intelligent simulation environment. A specific goal of the project is to assess the commercial feasibility of the IntelliSIM simulation environment. IntelliSIM is essentially several thousand lines of PROLOG and C code that run the state-of-the-art package SIMAN for the user. The user is guided through a new generation of simulation methodology which has considerably fewer abstractions required.

The emphasis is now on domain knowledge from the industrial setting and on the design questions at hand. A user-friendly intelligent simulation environment selects the

appropriate generic model from related knowledge bases through a user interactive expert system. The knowledge bases also contain and uses a group technology (GT) meta-code for user goals and the GT code for the physical system when selecting a model. User input generates a specific model by providing data, selecting experimental designs, running the executable model, and providing standard outputs. In this way a generic model becomes specific through user interaction. Plots and confidence intervals of performance measures are included when appropriate. This not only allows the user to model an appropriate system, but it also decreases a user's difficulties in modeling the system and in analyzing output results.

1.4. System Simulation Laboratory Publications

GUIDES Related Publications

JOURNAL ARTICLES / BOOK CHAPTERS

- Cochran, J.K. and Lin, L., "Assembly Line System Dynamic Behavior for High Priority Job Order Processing", *International Journal of Production Research* (1991 - to appear)
- Lin, L., Cochran, J.K., and Sarkis, J., "A Metamodel-Based Decision Support System for Shop Floor Production Control", *International Journal of Computers in Industry*, (1991 - to appear)
- Lin, L. and Cochran, J.K., "Estimating Simulation Metamodel Parameters for Unexpected Shop Floor Real Time Events", *Computers and Industrial Engineering* 19:1-4, pp. 62-66 (1991)
- Ozdemirel, N.E., Mackulak, G.T., and Cochran, J.K., "A Group Technology Classification and Coding Scheme for Discrete Manufacturing Simulation Models", *International Journal of Production Research* (1991 - accepted with revision)
- Lin, L. and Cochran, J.K., "Metamodels of Production Line Transient Behavior for Sudden Machine Breakdowns", *International Journal of Production Research* 28:10, pp. 1791-1806 (1990)
- Cochran, J.K. and Chang, J., "Optimization of Multivariate Simulation Output Models Using a Group Screening Method", *Computers and Industrial Engineering* 18:1, pp. 95-103 (1990)
- Mackulak, G.T. and Cochran, J.K., "Generic/Specific Modeling: An Improvement to CIM Simulation Techniques", in *Optimization of Manufacturing Systems Design*, D. Shunk editor, North-Holland, pp. 237-260 (1990)
- Cochran, J.K. and Cheng, C.S., "Automating the Procedure for Analyzing Univariate Statistics in Computer Simulation Contexts", *Transactions of the Society for Computer Simulation* 6:3, pp. 173-188 (1989)
- Hong, S.C., Cochran, J.K., and Mackulak, G.T., "Specification of an Architecture for Intelligent Simulation Environments", in *Simulation and AI*, W. Webster editor, SCS Series 20:3, pp. 77-82 (1989)

CONFERENCE PROCEEDINGS / PRESENTATIONS

- Cochran, J.K., Stahlman, E.J., Lin, L., and Mackulak, G.T., "The Evolution of Transient Behavior Modeling Techniques for High-Volume Manufacturing Processes", *Conference on Fossil, Chemical, and Manufacturing Simulation*, Newport Beach, CA (1992-to appear)
- Stahlman, E.J. and Cochran, J.K., "A Simulation-Based Approach to Dynamic Resource Allocation", *Twenty-Second Annual Pittsburgh Conference on Modeling and Simulation*, pp. 1065-1072, Pittsburgh, PA (1991)
- Cochran, J.K. and Gaafar, L.K., "An Integrated Simulation-Based Decision Support Tool for Shop-Floor Environments", *Conference on Modeling and Simulation on Microcomputers*, pp. 81-86, San Diego, CA (1990)
- Mackulak, G.T. and Cochran, J.K., "The Generic-Specific Modeling Approach: An Application of Artificial Intelligence to Simulation", *IEEE Integrated Systems Conference*, pp.82-87, San Antonio, TX (1990)
- Gaafar, L.K. and Cochran, J.K., "Developing a Real-Time Simulation Tool for Shop-Floor Decision Making", *SCS Summer Computer Simulation Conference*, pp. 79-85, Austin, TX (1989)
- Lin, C.Y. and Cochran, J.K., "An Automated Experimental Design Preprocessor for Unbiased Simulation Analyses", *SCS Summer Computer Simulation Conference*, pp. 86-90, Austin, TX (1989)
- Mackulak, G.T. and Cochran, J.K., "Generic/Specific Modeling: An Improvement to CIM Simulation Techniques", *International Conference on Modeling and Simulation for Optimization of Manufacturing Systems Design and Application*, Tempe, AZ (1989)
- Ozdemirel, N.E., Mackulak, G.T., and Cochran, J.K., "A Preliminary Group-Technology Classification Scheme for Manufacturing Simulation Models", *SCS Winter Simulation Conference*, pp. 697-703, San Diego, CA (1988)

SIGNIFICANT RESEARCH REPORTS

- "Intelligent Simulation Systems for Automated Factories", Six Progress Reports for the **Institute for Manufacturing and Automation Research** by J. Cochran, G. Mackulak and M. Melkanoff (54, 28, 24, 73, 77, and 38 pp), December 1991, June 1991, December 1990, May 1990, December 1989, and May 1989
- "GUIDES: A Generic User-friendly Intelligent Design Environment for Simulation", Final Report for the **National Science Foundation I/UCRC** for Manufacturing and Automation by J. Cochran and G. Mackulak, Volumes 1 and 2 (245 and 195 pp), August 1990

MASCOT RELATED PUBLICATIONS

JOURNAL ARTICLES / BOOK CHAPTERS

- Mackulak, G.T. and Cochran, J.K., "Mascot: A Prolog-Based Simulation Modeling and Training Environment", in *Modeling and Simulation Methodology: Knowledge Systems Paradigms*, Elzas, Oren, and Zeigler editors, North Holland, Elsevier Science Publishers, pp. 145-160 (1989)
- Cochran, J.K. and Mackulak, G.T., "Low Cost Artificial Intelligence Can Aid Design and Simulation", *CIM Review* 3:3, pp. 33-36 (1987)

Lin, L. and Cochran, J.K., "Optimization of a Complex Flow Line for Printed Circuit Board Fabrication by Computer Simulation", *Journal of Manufacturing Systems* 6:1, pp. 47-57 (1987)

CONFERENCE PROCEEDINGS / PRESENTATIONS

Castillo, D. and Cochran, J.K., "A Comprehensive Microcomputer-Based Statistical Program for Analyzing Simulation Input and Output Data", *Conference on Modeling and Simulation on Microcomputers*, pp. 3-11, San Diego, CA (1987)

Cochran, J.K., Mackulak, G.T., Castillo, D. and Du, E., "Configuring Available Software Into an AI/ES Environment for Automated Manufacturing Simulation Design on the PC", *Conference on Simulation of Computer Integrated Manufacturing Systems and Robotics*, pp. 1-7, San Diego, CA (1987)

Mackulak, G.T. and Cochran, J.K., "MASCOT: A Prolog-Based Simulation Modeling and Training Environment", *International Symposium on Modeling and Simulation Methodology - Intelligent Environments and Goal-Directed Models*, pp. 18-21, Tucson, AZ (1987)

SIGNIFICANT RESEARCH REPORTS

"Generic System Simulation Development on the PC (Phase I-III)", Final Report for **Motorola Management Institute** by G. Mackulak, J. Cochran and D. Shunk, Volumes 1, 2, 2B, and 3 (21, 115, 257, and 240 pp), August 1987

OTHER SSL PUBLICATIONS

JOURNAL ARTICLES / BOOK CHAPTERS

Cochran, J.K. and Chen, M.T., "An Integrated MultiComputer DSS Design for Transport Planning Using Embedded Computer Simulation and Database Tools", *Decision Support Systems* 7:1, pp. 87-97 (1991)

Cochran, J.K. and Paul, B.K., "QUAL: A Microcomputer System for Qualitative Simulation", *Simulation* 54:5, pp.300-308 (1990)

Dessouky, Y. and Mackulak, G.T., "Parallel Unrelated machine Scheduling to Minimize Makespan with Changeover", *Journal of Manufacturing Systems* 9:2, pp. 139-150 (1990).

Paul, B.K. and Cochran, J.K., "A Microcomputer Tool for Qualitative Simulation Based on an Object-Oriented, Device-Centered Ontology", in *AI and Simulation Theory and Applications*, W. Webster and R. Uttamsingh editors, SCS Series 22:3, pp. 22-27 (1990)

Cochran, J.K. and Lin, L., "An Application of Computer Simulation to Freight Transport Systems", *Journal of the Operational Research Society* 40:5, pp. 433-441 (1989)

Radwan, A.E., Cochran, J.K., and Farris, M., "A Decision Support System For Freight Transport In Arizona", *Transportation Research Record* 1179, pp. 23-27 (1989)

Glenney, N. and Mackulak, G.T., "Modeling and Simulation Provide Key to CIM Implementation Philosophy", *Simulation Modeling Manufacturing & Service Systems*, IEE Press, pp. 58-67 (January 1988)

- Cochran, J.K., "Techniques for Ascertaining the Validity of Large - Scale Production Simulation Models", *International Journal of Production Research* 25:2, pp. 233-244 (1987)
- Cochran, J.K., Hubele, N.F., and Sykes, D.J., "A New Generation of Flight Simulators: Design Configuration With Discrete-Event Simulation", in *Simulators IV*, B. Fairchild editor, SCS Series 18:4, pp. 234-238 (1987)

CONFERENCE PROCEEDINGS / PRESENTATIONS

- Cochran, J.K., Crowe, T.J., and Ogle, M.K., "A Generic Simulation Experiment for Evaluating Dispatching Rules in Flexible Manufacturing Systems", *Twenty-Second Annual Pittsburgh Conference on Modeling and Simulation*, pp. 1452-1458, Pittsburgh, PA (1991)
- Savory, P.A., Mackulak, G.T., and Cochran, J.K., "Material Handling in a Six Machine Flexible Manufacturing System Processing Part Families, *SCS Winter Simulation Conference*, Phoenix, AZ (1991)
- Crowe, T.J., Cochran, J.K., and Stahlman, E.J., "Design Features of a CIM Simulation Planning Tool Based on a C Language Coroutine Implementation Strategy", *National ORSA / TIMS Conference*, Las Vegas, NV (1990)
- Betz, M.J. and Cochran, J.K., "The Advantages of Network Analysis in Modeling Global Goods Movement", *National Research Council Annual Meeting of Transportation Research Board*, Washington, DC (1988)
- Sykes, D.J., Cochran, J.K., and Young, H.H., "Development of Diagnostic Expert Systems Using Qualitative Simulation", *Conference on Artificial Intelligence and Simulation*, pp. 32-38, San Diego, CA (1988)
- Yang, Y.C., Liu, M.C. and Cochran, J.K., "Knowledge Based Scheduling for Flexible Manufacturing Systems", *International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Vol. 2, pp. 1075-1082, Tullahoma, TN (1988)
- Cochran, J.K., "Intelligent Simulation Versus Simulating Intelligence", *Conference on Philosophical Aspects of Artificial Intelligence*, Tempe, AZ (1986)

THESIS/DISSERTATION

MASTERS THESIS

- Cole, Vera (1989) "An Object-Oriented Paradigm For Incremental Manufacturing Automation and Redistribution of Developmental Responsibilities", Major Professor: Dr. Jeffery Cochran
- Du, Eric (1986) "Development and Evaluation of a Graphical Planning/Layout Pre-Processor for Flexible Manufacturing Systems", Major Professor: Dr. Gerald Mackulak
- Gaafar, Lotfi K. (1989) "An Integrated Shop-Floor Simulation based Decision Support Tool", Major Professor: Dr. Jeffery Cochran.
- Lin, Chin-Yuh (1988) "Real-time Automatic Output Analysis" [Engineering Report], Major Professor: Dr. Jeffery Cochran.
- Paul, Brian K. (1988) "A Decision Support Heuristic Based on Qualitative Simulation for Projecting Organization Acceptance of Computer Integrated Manufacturing", Major Professor: Dr. Jeffery Cochran

Snyder, Jordan (1989) “The Decide Node: A Nonmonotonic Reasoning System Applied to Discrete Event Simulation”, Major Professor: Dr. Gerald Mackulak

Yang, Ying-Che (1988) “Knowledge-Based Scheduling in Flexible Manufacturing Systems”, Major Professor: Dr. Jeffery Cochran

DOCTORAL DISSERTATIONS

Hong, Suck-Chul (1991) “Relational Database Implementation for Computer Simulation Results Over the Life Cycle of Simulation Models in CIM”, Major Professor: Dr. Jeffery Cochran

Lin, Li (1989) “Metamodels of Assembly Line System Transient Behavior for Selecting Dynamic Events”, Major Professor: Dr. Jeffery Cochran.

Ozdemirel, Nur E. (1990) “Generic Manufacturing Simulation Model Construction Based on Group Technology Classification of Models”, Major Professor: Dr. Gerald Mackulak

Stahlman, Edward J. (1991) “A Simulation Methodology for Dynamic Event Characterization In Discrete Manufacturing”, Major Professor: Dr. Jeffery Cochran

CHAPTER 2

RESEARCH CONCEPTS IN INTELLISIM

2.1. Introduction

This section of the user's manual contains a theoretical and program description of IntelliSIM (**Intelligent Simulation**). Included is a survey of simulation concepts, and a description of the capabilities implemented in this package. Readers of this section will include simulationists, programmers, IntelliSIM support personnel, and anyone interested in how the program works. References are included in Section 2.8 so that the interested reader may find additional background material.

This section is divided into four general areas. The first, sections 2.1 to 2.4, provide general background information including the impetus for IntelliSIM and a discussion on intelligent simulators. Section 2.5 is the theoretical basis of IntelliSIM and the heart of this section. It discusses the automated capabilities and provides a description of the Group Technology (GT) scheme embedded within IntelliSIM. The remainder of the section contains relevant information regarding the implementation of IntelliSIM.

2.1.1. Expert System Simulation

Computer simulation is a tool that enables evaluation of various manufacturing process scenarios without developing a physical system or interrupting an existing process. It has a relatively long history of use and some significant successes (CIM Review, Vol. 3, Winter 1987, provides an in-depth analysis of simulation modeling for the manufacturing environment.) For an expert simulation system to succeed, it must contain rules and heuristics that assist in the effectual resolution of both planning and design issues (Cochran and Mackulak 1987.)

Computer simulation is largely concerned with mathematical or logical modeling and therefore requires skills not widely available or easily retained from training. One

common reason why engineers fail to meet design goals with simulation is that they have not received, or retained, basic statistical system simulation analysis training. To compensate for these missing skills, such expertise must be built into the simulation environment.

2.1.2. Model Creation

One of the more difficult aspects of a simulation study is building a model from scratch. It involves many complex steps, such as defining the system, the level of aggregation or disaggregation required, the statistics to be collected, deciding what elements to include, what simplifying assumptions are valid, and how interactions occur. One important goal of any expert simulation environment is to assist as much as possible in model building.

The mechanics of model building have been simplified greatly by the development of network symbol-based simulation languages and systems that help to build a model from the basic symbol icons through the use of a light pen or mouse. Nevertheless, modelers must still acquire the important abstraction techniques by example, consultant, or education courses.

A useful tool developed to overcome these difficulties is the generic-specific model. The term generic-specific appears to be an oxymoron, but it best describes the process wherein a user selects (with AI guidance) a generic simulation format and fine-tunes it to solve specific problems. This process avoids many failures in model creation experienced by novice simulation modelers and decreases the amount of effort required to build a working model. The advantages to this generic-specific approach are that it:

- simplifies model modification,
- provides organizing concepts for the data base structure that maintains models,
- allows simple implementation of generic-specific statistics collection, reporting, and graphical displays, and
- allows segmented file creation based on user needs.

A complete description of generic-specific modeling is given in Section 2.6, Fundamental Concepts of Generic/ Specific Modeling.

2.1.3. A Microcomputer Based Approach

Expert systems and generic-specific models can be implemented on all levels of computer platforms: microcomputer, workstation, mini-computer, or mainframe computer. Through an analysis of the advantages and disadvantages of each level we have decided on a microcomputer based approach.

The primary advantage of microcomputers is their low cost. When the total cost of hardware and software does not exceed \$10,000, dozens of stations can be funded for the cost of one LISP machine station, or a mainframe system. Another advantage of this approach is robustness. By this we mean that a PC can be put in an office, on a shop floor and, in some cases, is portable. It is easy to maintain, repair, and upgrade. If a microcomputer is stolen or destroyed, it represents less of an incremental loss, not only in terms of replacement cost and insurance premiums, but also in terms of disruption of project schedules. The final advantage we will mention here is ease of transition. The user is probably more familiar with microcomputers due to their availability. This application is just another set of programs which happens to be particularly user-friendly. The local communication network requires no modification, and interested personnel can play with the source code if they like.

One practical requirement of an expert system simulation environment is that it respond to the user's computing needs in a reasonable amount of time, where response time includes overall study completion time. Clearly, the execution time on a microcomputer will be longer than the corresponding execution time on a mainframe computer. Since the difference in execution time will vary based on the simulation model and the extent of the analysis, the significance of this weakness will vary from project to project. Response time is the one clear disadvantage of the microcomputer.

2.1.4. Planning and Design Issues

The planning and design functions associated with incorporation of automation technology are of paramount importance. They are as important as the statistical issues. Industry is full of cases of high technology system development and design that have resulted in ineffective, inappropriate, and incorrect technology applications. Most of these failures can be traced to the simple concept of poor planning prior to system development. Manufacturers are especially susceptible to this problem since most manufacturers are faced with constant product updates, engineering changes, design staff turnover, and new product development strategies necessary to remain in business in today's highly competitive market.

A key to successful system development, design and incorporation is, therefore, advance planning and design. Effective planning must be accomplished prior to considering details pertaining to equipment type, costs, or facility design.

The problem is that most engineers are ill-equipped to perform effective planning. Frankly, many engineers have no concept of how planning should occur to integrate the various components of a high technology environment. Although this is a high-priority area, few tools are available to apply AI/ES techniques for satisfying the planning goal. It appears that CIM is providing a motivation for development of both computer-aided planning and design. These requirements are not new; it is just that automated manufacturing has experienced more than its share of integrated system development and design failures and is now willing to pay for research and development along these lines.

This emphasis on intelligent planning and design dictates the need for maximizing what little expertise is available from the prior successes. That shortage of expertise is based, in part, on the fact that CIM planning and design have additional goals and motivations than traditional activities: the scope is larger, the organizational infrastructures are more complex, and the time frame is longer. A company, in order to survive, must move to a higher level of technology even though such change may not

bring short-term pay back. CIM must account for the issue of survival and justify new technologies or design adaptations of existing technologies where long-term, nonquantitative benefits may overshadow short-run profit concerns. Any decision support system in this context has great potential requirements for AI/ES.

Nevertheless, we are not suggesting that the basic issues of facilities layout, scheduling, and process and material control are any less important than before. Rather, automated manufacturing in today's markets increases the burden on engineers and managers. The project team needs to evaluate many diverse alternatives at early stages in a project so as not to exclude potential solutions, regardless of how infeasible or far-fetched they appear based on traditional experience. As the project takes shape, a clearer direction will emerge and non-feasible alternatives can be eliminated. Until this stage, the design team must refrain from making costly and time-consuming detailed analyses of the different proposed alternatives.

The integration of generic-specific modeling and expert systems in an Intelligent Simulation Environment will provide the skills required for managers and engineers to effectively use simulation throughout the planning and design phases of a project. The users will not need to become simulation or statistical experts, models will be able to be built and executed quickly, and the integrity of simulation results will be increased. Thus, simulation will become more useful to nonsimulationists since it will be easy to apply and will provide timely results.

2.2. Discrete-Event Simulation

Simulation models can be divided into a number of categories. This project deals specifically with discrete-event simulation. This section provides an overview of discrete-event simulation and how expert systems are being used to augment current simulation techniques.

2.2.1. Steps in a Simulation Study

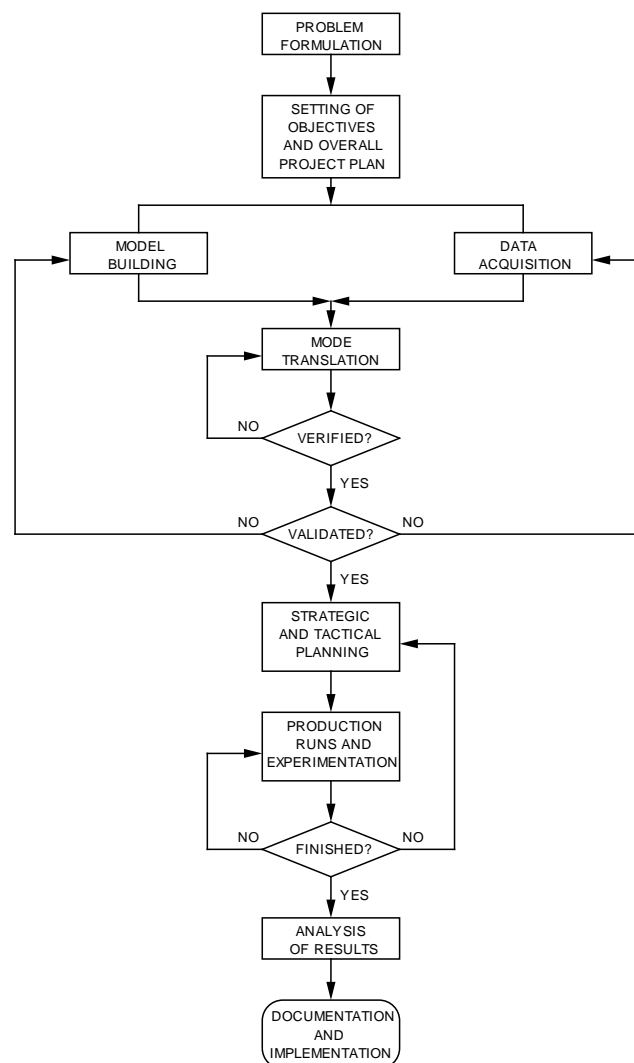


Figure 2.1: Steps in A Simulation Study

Figure 2.1 shows a high level flowchart of the steps required of a simulation practitioner (Mackulak, et al. 1987.) Three central tasks are: 1) define the experiment, 2) simulate the alternatives, and 3) process the output (provide output analysis of a single system and provide the capability to compare multiple systems.) The subtasks involved in each task are:

A. Define the experiment

1. Define the performance measures and statistics to be collected.
2. Determine what level of precision to use for hypotheses.
3. Try to find an existing model to modify or hire a consultant.
4. Determine the appropriate data collection points during the simulation.
5. Decide whether a single long run or multiple runs are appropriate.

B. Simulate the alternatives

1. Execute the simulation based on the experiment developed during the question and answer session.
2. Collect appropriate data at specified data collection points.
3. After data has been collected, display a message to the user which gives the current status of the experiment. Let the user know approximately how many more runs and how much more time is required to achieve the desired goals set down by the user. Give the user the opportunity to continue, readjust, or terminate, saving all data generated up to this point.
4. Perform tests for autocorrelation on data collected. Interrogate a knowledge base to determine if too much autocorrelation exists. If there is evidence of a high amount of autocorrelation, then the knowledge base should offer advice on reducing autocorrelation.

C. Process the output

1. Capability to test data for autocorrelation.
2. Capability to truncate data if necessary to reduce start-up bias.
3. Perform confidence interval calculations.
4. Provide test for normality to determine if parametric procedures can be applied to the data.
5. Provide capability to compare alternative systems using both parametric and nonparametric procedures.
6. Use graphics to display statistical information.
7. Perform descriptive statistics on sample data.

2.2.2. Classification of Simulation Tools

Simulation tools can be categorized based on the amount of modeling and programming required by the user (Crowe et al. 1990.) Six levels of categories are presented in Figure 2.2, based on a five level taxonomy presented by Schmidt (1980).

In the lowest class are high-level programming languages. Performing a simulation study with only a high-level, general-purpose programming language not only requires the user to develop his own modeling paradigm, but also requires him to program every aspect including event scheduling, list maintenance, flow control, data dimensioning and maintenance, and statistics gathering.

The second class of simulation tools are low-level simulators. Low-level simulators provide the user with flow control, generation of random numbers, and evaluation and presentation of results. Schmidt points out that with a low-level simulation tool the user must construct the simulation model from the tool's available functions and features. With the low-level simulator the user is still required to perform the modeling on his own, but he receives some assistance in the programming phase.

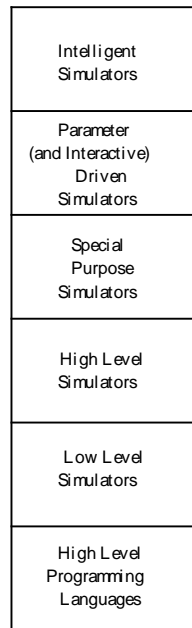


Figure 2.2: An Extension to Schmidt's [1980] Taxonomy of Simulation Tool
(Crowe et al. 1990)

The third class of simulation tools are high-level simulators. High-level simulation tools contain all the features of low-level simulators with the addition of special constructs for quick and easy implementation of models. With high-level simulation tools the user must perform the modeling on his own but is aided by having a modeling paradigm to guide him. At this level the user is still responsible for all programming responsibilities. Almost exclusively, high-level simulation tools are implemented as simulation languages.

The fourth class of simulation tools are called special-purpose simulators. Special-purpose simulation tools are directed at relatively small specific application areas and contain constructs which are only useful for modeling those applications. Again at this level the user is responsible, although guided, for all modeling and programming activities.

The fifth class of simulation tools are called parameter-driven simulators. With parameter-driven simulation tools the structure of the model to be simulated is specified through a stream of input data. Although the user must still perform the modeling, he is relieved of all programming. A new type of simulation tool, generally called interactive, has emerged which could be grouped as a sub-class of Schmidt's parameter-driven tools. When modeling with an interactive-simulation tool, the user specifies his model step-by-step interactively, rather than through a parameter type table.

Recently, a sixth level to Schmidt's taxonomy has emerged, intelligent simulators. With intelligent simulators the user needs only to describe the system to be modeled, as in the parameter-driven simulators, and the goals of the simulation. The intelligent simulator will then perform the necessary tasks required to meet the user's goals and present the user with the results. Thus, the user is relieved of the task of modeling as well as programming. The next section provides an in depth discussion of the requirements and capabilities of this level of simulation.

2.3. The “Perfect” Simulation Environment

As you may have deduced by now, we have definite ideas on the features required in a "perfect" simulation environment. It must be an expert in significant planning and design issues. This alone leads naturally to the synthesis of ES/ SIM which so many other authors have emphasized. Reddy, et al. (1986) have provided one of the best lists of features that the perfect system would have. Their emphasis is on knowledge-based simulation rather than the approach we prefer, so our list is shorter than theirs (as we will discuss in the next section), but the intent is the same. Hendrikson (1983) also has an excellent discussion to which we refer the interested reader. In particular, he discusses design philosophy, as well as practicum, which lends a comprehensive flavor to his discussion.

At any rate, Table 2.1 is our list of key features which must be present. Most of these have received attention in the literature before. We present them in no particular order, nor is the list intended to be comprehensive. First, we turn our attention to what, at first glance, may appear to be a radical concept. We propose that there is an essential feature: the use of a microcomputer.

Central to our approach to constructing an AI/ ES/ SIM/ CIM environment is the use of current simulation tools and microcomputers. However, the prominent trend in current efforts to construct such environments is in the direction of rule-based object-oriented simulation systems. Adelberger (1986) has surveyed this area.

There is no doubt that there is a definite efficiency to be gained by using special architecture hardware. Furthermore, and more significantly, the ability to implement powerful "intelligent" constructs is easier by virtue of the dedicated languages which have been, and are being, developed for these machines. An approach with much merit are the projects aimed at writing new simulation languages (number-crunching, event calendar, and all) for these machines or their close relatives.

Table 2.1: Fifteen Features of the Perfect Simulation Environment

1	No user coding required
2	Interactive access available via windows and menus
3	Contains preexisting models to modify rather than to build from scratch
4	Contains model verification capabilities with execution tracing and display
5	Provides automated input statistics analysis
6	Provides automated model execution
7	Provides automated output statistics analysis
8	Contains natural execution break points to interrupt sessions
9	Provides generous graphics display ability
10	Has built-in word processing for report generation
11	Has rapid execution or an ability to execute experiments in a background environment
12	Provides experimental design capability for scenario comparisons
13	Provides for pleasant user-screen interaction
14	Contains built-in instruction and training
15	Uses a microcomputer

These knowledge-based simulation (KBS) approaches share many features with current generation general purpose simulation (GPS) languages (for example, SIMAN and SLAM II), but have a number of significant differences which make them good candidates for the "perfect" environment. Shannon, et al. (1985) is a good reference for a clear and thorough discussion of the differences. Ben-Arieh (1986) focuses also on the CIM context in which we are interested here. Briefly, the differences are as follows:

- (1) GPS is based principally on numeric-based computation while KBS is based principally on logic-based computation.
- (2) GPS requires the user to test data assumptions, perform output analysis, design the experiment, and so on, while KBS includes that logic.
- (3) GPS relies on algorithms determined in advance of execution, while KBS uses more dynamic pattern-invoked processing.
- (4) GPS tends to combine data, knowledge, and control in its programming, while KBS treats them as distinct modules.

These differences, frankly, represent improvements over the current simulation tools.

Must we then abandon (or at least rewrite) the advantages contained within general purpose simulation languages? Not at all. We, and a handful of other researchers, prefer to execute those languages through an AI/ ES interface. The AI/ ES interface is responsible for the bulk of the logic-based computation, while the current simulation language does the numeric-based computation. To expand on how such an interface can be used to create the intelligent simulation environment outlined above, the next section discusses specific issues concerning implementation.

2.4. Critical Configuration Issues

2.4.1. The Topology of Expert Simulation

In his interesting article, O'Keefe (1985) proposes a taxonomy of topologies for combining expert systems and simulation. These topologies are labeled as:

- (1) Embedded (ES within SIM or SIM within ES)
- (2) Parallel (accessing each other independently)
- (3) Cooperative (sharing some code)

It can be argued that some of the groups may be combined or refined, but O'Keefe's basic notion is important.

In the context of this project, we must decide or discover which topology we will use when configuring software. It is straightforward to know which program "calls" the other one, but the issue is more subtle than that.

We wish to bring up this point here as something to be aware of when using the microcomputer configuration approach to creating an AI/ ES/ SIM environment. The choice may have an impact on future generations and refinements of an environment. It may, on the other hand, be caused by lack of access to source code; that is, where one program simply runs the other by providing it input (perhaps as a file or passed as a parameter) and then using the output.

For this environment, the ideal topology is a parallel implementation. A parallel environment is not only faster, but is also more flexible since the user could be using the computer for multiple tasks at one time. This would be especially useful when very long simulation experiments are being executed; the simulation experiment could be executed in the background while the user executes other programs, such as word processing, in the foreground.

Nevertheless, no widely available software is available to implement a parallel topology within our guidelines. Therefore, we use an embedded topology which is more practical and easier to implement.

2.4.2. Taxonomical approaches to Simulation

Classification of simulation models is not a new idea. In fact, various simulation taxonomies available in the literature are commonly known and already included in many simulation textbooks. Among them, perhaps the most elementary classification is discrete simulation versus continuous simulation, which is based on how the values of the system state variables change. Other basic classifications are static versus dynamic simulation, and deterministic versus stochastic simulation.

Additional classification criteria can be based on the following characteristics of simulation models: purpose of simulation, model characteristics, relationships among entities, attribute characteristics, and variable characteristics. Many other classifications can also be made based on the user's needs and goals.

The above well-known classification schemes are useful in documentation and training, but they offer limited information regarding the simulation model structure. They do not, for example, tell what the basic simulation entities are for a particular domain, how those entities interact, and what kind of material and information flow characteristics are involved. They also lack information regarding the structure of simulation routines and how those routines should be interfaced.

The constructed models must satisfy specific user requirements or goals. These models will allow creation of simulation scenarios without user abstraction, a skill that has been found lacking in non-expert modelers. To achieve these objectives, the knowledge regarding model structures must be built into the model base. We have chosen to define this knowledge through specification of a classification scheme. This classification scheme will constitute a framework for the development of such a model base, and will be based on the structural properties of simulation models.

2.5 Automation of Simulation Model Development

As a result of recent developments in applied artificial intelligence and knowledge based systems (KBS), many researchers from other disciplines focused their attention on the potential use of AI/ ES techniques in their areas. KBS can assist simulationists in many ways, such as model construction, input preparation, and output analysis (designing and performing statistical experiments.) The emphasis of this research is to build a generic simulation model base, and to assist the user in developing the appropriate specific model for his or her purposes through interaction with the model base.

2.5.1. The GT Scheme

Although computer simulation is a well-studied and highly structured research area, its use in industry is still limited due to the fact that simulation modeling requires expert knowledge. This expertise must be embedded in an integrated simulation software to provide non-expert user access to this valuable tool. Simulation experts know very well that there are similarities between the models they build for different manufacturing systems. In this section, a group technology (GT) classification scheme for the discrete manufacturing simulation models, based on these similarities, is discussed. The GT scheme will serve as a basis for creating a model base consisting of generic models within the framework of an intelligent simulation environment. Specific user models will be constructed by modifying and configuring the generic models rather than building them from scratch, reducing the effort expected of the user in the process of model building.

2.5.1.1. GT Scheme Form

In developing the GT code, a "digit" is required to represent each common property, and the collection of all digits constitute the coding scheme. Each digit has at least two levels. The two levels of a digit may simply represent existence or nonexistence of a certain property in the model. A digit may have more than two levels, each indicating a different option for the corresponding model property (Ozdemirel 1990.)

The GT code is a hybrid type. Some digits are always used, and interpretation of the digit in the sequence is fixed, independent of the preceding digits (polycode or chain-type structure). Some others are used only when certain levels of the previous digit are chosen. When used, interpretation of the digit depends on the value of the preceding digits (monocode or hierarchical structure). The hybrid structure is defined by the nature of the model properties. For example, dispatching rules to decide which job to process next is a property that needs to be considered as independent from the rest of the properties, implying a polycode structure. The decision regarding what happens to balking jobs, on the other hand, has to be made only when the work-in-process areas are assumed to have limited capacity, and requires a monocode structure.

Digits of the GT code are classified into two major groups: user goals of simulation, and physical system characteristics. Each group of digits is organized into a tree structure to obtain a multi-level coding scheme according to the group technology classification approach.

A complete specification of the GT Code Scheme implemented in IntelliSIM can be found in Ozdemirel (1990).

2.5.1.2. Generic Model Structures

A generic model can be broken down into five major sections based on the grouping of GT digits and clusters generated for each digit group. Interchangeable modules are built for each of the sections to generate various generic models. The five model sections are independent of each other, so that like model sections can replace each other without any changes in the rest of the model sections. The model sections are:

- a job arrival module in which entities representing batches of different job types are created and entity attributes are initialized,
- a station entry module in which jobs arrive at a work station, release material handling equipment and pallets if they are used, and are routed to the correct station type according to the operation sequence specified for the job type,

- a station processing module in which jobs are processed at one of the various station types defined in the GT scheme,
- a find next station module in which certain actions are taken before the jobs are routed to their next destination using certain types of material handling equipment, and
- an equipment breakdown and scheduled maintenance module in which breakdown and maintenance activities are modeled, if required.

There are fourteen modules in total. The modules are given in Table 2.2, grouped by the model sections for which they are built. In the second column of the table, a key GT digit is defined for each of the five module groups. The value of the key GT digit for a module draws the distinction between the modules within a group. A complete generic model is generated by choosing a module from each group and coupling them. The key GT digits define the restrictions on the choice of modules. For example, if the two module groups are assigned the same key GT digit, then only those modules that carry the same value of the key digit can be coupled.

2.5.1.3. Naming Generic Model files

The key GT digits that are assigned to modules are also used in naming the generic model files. The first letter in the file name of a generic model is always a "G", indicating that it is generic. The next two characters are either "Q0" or "Q1", representing the configuration of WIP storage areas. "Q0" means that there is one WIP area per station for incoming jobs ($Q_3 = 0$), and "Q1" is used when there are two separate WIP areas per station for incoming and outgoing jobs ($Q_3 = 1$.) The last option for this digit, one WIP area per station for both incoming and outgoing jobs ($Q_3 = 2$), is ignored because it is found in very few studies in the review and GT coding process.

The fourth character in the file names is an "A" for the job arrival pattern, and the fifth one takes the value of J_1 , indicating one of the following four options for creating job arrivals.

Table 2.2: Generic Modules

Module Group Module Name Names	Key GT Digits	Generic	Model
<u>Job arrival modules</u>			
ARRVL0	$J_1 = 0$	GQ*A0S**.MOD	
ARRVL1	$J_1 = 1$	GQ*A1S**.MOD	
ARRVL2	$J_1 = 2$	GQ*A2S**.MOD	
ARRVL3	$J_1 = 3$	GQ*A3S**.MOD	
<u>Station entry modules</u>			
ENTRY0	$Q_3 = 0$	GQ0A*S**.MOD	
ENTRY1	$Q_3 = 1$	GQ1A*S**.MOD	
<u>Station processing modules</u>			
PROCES99	$M_{31}, A_{31} = 9$ and $M_{32}, A_{32} = 9$	GQ*A*.MOD	
PROCES00	$M_{31}, A_{31} = 0$ and $M_{32}, A_{32} = 0$	GQ*A*S00.MOD	
PROCES02	$M_{31}, A_{31} = 0$ and $M_{32}, A_{32} = 2$	GQ*A*S02.MOD	
PROCES12	$M_{31}, A_{31} = 1$ and $M_{32}, A_{32} = 2$	GQ*A*S12.MOD	
PROCES04	$M_{31}, A_{31} = 0$ and $M_{32}, A_{32} = 4$	GQ*A*S04.MOD	
<u>Find next station modules</u>			
FNDNXT0	$Q_3 = 0$	GQ0A*S**.MOD	
FNDNXT1	$Q_3 = 1$	GQ1A*S**.MOD	
<u>Equipment breakdown and scheduled maintenance modules</u>			
BDOWN	$B_1 = 3, B_2 = 2, B_3 = 0, B_4 = 0$	GQ*A*S**.MOD	

A0 - All jobs are available at the beginning of the simulation period.

A1 - Job types are randomly specified as percentages of random arrivals.

A2 - Interarrival time distributions are specified for each job type.

A3 - Deterministic arrival times are specified for each batch of each job type.

The last three characters in a model name are about the usage pattern of the optional resources for machining and assembly stations. Inspection and fixture loading/unloading stations are ignored in this respect, because it is found in the review of studies that these processes do not need to be broken down into three suboperations. Machine setup, machine loading/unloading, and processing are either not separable for inspection stations or, if they are, optional resources such as inspection operators are utilized during the entire operation. Fixture loading/unloading takes much less time compared to machining or assembly of jobs, and suboperations are not valid for this process type.

For machining and assembly stations, the GT scheme allows various usage patterns for optional resources. Only the following five combinations, however, are observed in the studies reviewed, and included in the generic models.

Last three characters are not used - No optional resources are used.

S00 - Both optional resource types are used simultaneously during the entire operation.

S02 - First optional resource type is used for the entire operation, second for machine loading/unloading.

S12 - First optional resource type is used for machine setup, second for machine loading/unloading.

S04 - First optional resource type is used for the entire operation, second for machine setup and machine loading/unloading.

Many studies make use of one optional resource type per station, and the second resource is not needed. Although generic model structures incorporate two optional resource types, one does not have to use both of them. For a particular station in a model, any of the two optional resources can be used, and the other may be ignored. For example, if one optional resource is needed for machine loading/unloading only, one of

the generic models GQ*A*S02.MOD or GQ*A*S12.MOD must be selected, and the second optional resource must be used. The choice regarding the optional resource in use is made in the experimental frame which is discussed in the following section.

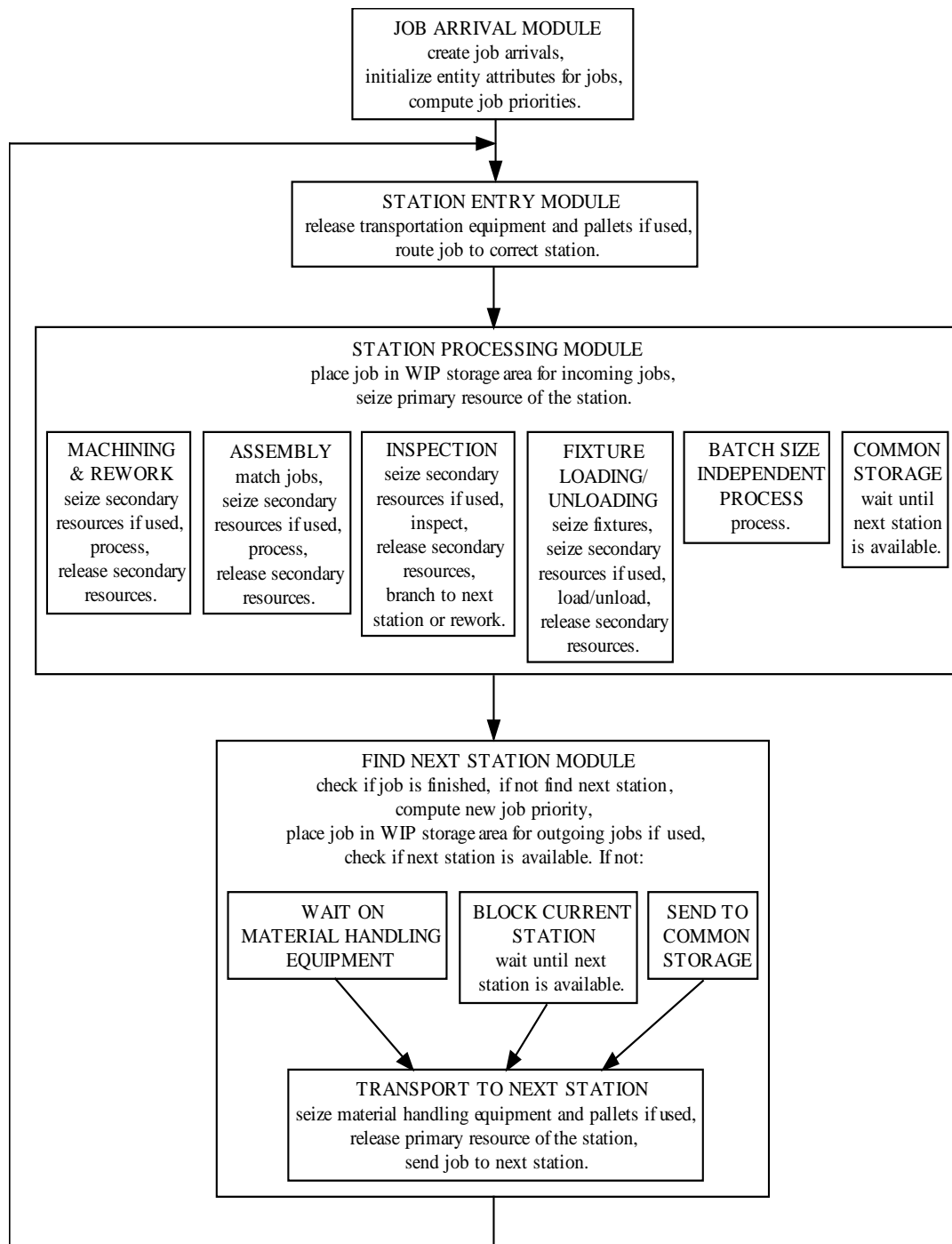


Figure 2.3: Flow of Jobs Through a Generic Model

Among the groups of modules in Table 2.2, only the station entry modules and find next station modules are dependent, having the same key GT digit. This is mainly because modeling of job processing based on the work stations approach is cyclic. Jobs leaving a station seize the material handling equipment in the find next station module. They cycle back to the station entry module for the next operation and have to release the material handling equipment before the processing starts. The general flow of jobs through a generic model and cycling property of work stations are schematized in Figure 2.3. Equipment breakdown and scheduled maintenance module is not shown in the figure because this module does not involve flow of jobs. Breakdown module operates independently, in parallel with the rest of the modules.

Feasible combinations of these modules generate forty generic models: two station entry/ find next station modules can be coupled with any of the five station processing modules, yielding ten basic models, and each of the four job arrival modules can be used to create job entities for each of these ten models, resulting in a total of forty combinations.

To select a generic model structure for a particular application, only the key GT digits need to be known. The rest of the GT code for the application is determined as the selected model is configured, and is stored in the experimental frame that is generated for the specific model.

2.5.2. Automated Analysis of Univariate Statistics

2.5.2.1. Traditional Input Data Analysis

Traditionally, there are four steps in developing a valid model of input data (Banks and Carson, 1984):

- (1) Data collection
- (2) Identification of underlying statistical distribution: this includes histograms and probability plotting. From the frequency distribution, a distributional

assumption is made. In some situations, prior information might be helpful from which to select a distribution or at least rule out inappropriate distributions.

- (3) **Parameter(s) estimation:** The parameter(s) which characterize the hypothesized distribution are estimated. The most commonly used tool is maximum-likelihood estimation.
- (4) **Goodness-of-fit tests:** The distributional assumption and the associated parameter estimates are tested for goodness of fit. If the null hypothesis that the data follow the distributional assumption fails, then the analyst returns to the second step, makes a different distributional assumption, and repeats the procedure.

Among the above four steps, specifying hypothesized distributions requires intensive human interpretation and is often subject to bias. It is frequently this activity that determines the ultimate success of input data analysis. Also, this process does not distinguish between distributions of roughly the same fit, in terms of the run-time of computer simulation models in which they will be placed (Cochran and Cheng 1989.)

2.5.2.2. Parameterization of Distributions

For a given distribution, there are several alternative ways to define or characterize the probability functions. Most parameters used to define the distribution function can be divided into the following basic types: location, dispersion and shape parameters (Pettitt and Stephens, 1977):

1. **Location parameters** - statistics that define the location of a distribution (mean, median, mode, quantiles.)
2. **Dispersion parameters** - statistics that characterize the variability of a distribution (standard deviation, variance, range, coefficient of variation.)
3. **Shape parameters** - statistics that characterize the deviation of a distribution from the normal distribution (skewness, kurtosis.)

Location and dispersion parameters are well-known in the published literature. For example, Hasting and Peacock (1975) provide a concise but complete description about the properties of each distribution function.

2.5.2.3. Automated Analysis Methodology

To avoid the drawbacks of subjective human-directed input data analysis, a new approach is presented. The goal of the approach is to automatically perform input data analysis and suggest appropriate distribution function(s.) The main features include automatic hypothesis generation, goodness-of-fit testing, and a final rule-based system to suggest appropriate distribution function(s.) Figure 2.4 depicts the methodology. The details are described as follows:

1. Compute the empirical skewness and kurtosis coefficient from input data.
2. Compute the deviation (distance) δ_j from theoretical distribution j on the (β_1, β_2) surface defined in Figure 2.5. A two-dimensional Euclidean distance is adopted to measure deviation. The distance from point $P_1 = (x_1, y_1)$ to point $P_2 = (x_2, y_2)$ is defined as:

$$\delta = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

On the other hand, the distance from point $P = (x_0, y_0)$ to line $ax + by - c = 0$ is defined as:

$$\delta = \left| \frac{ax_0 + by_0 - c}{\sqrt{a^2 + b^2}} \right|$$

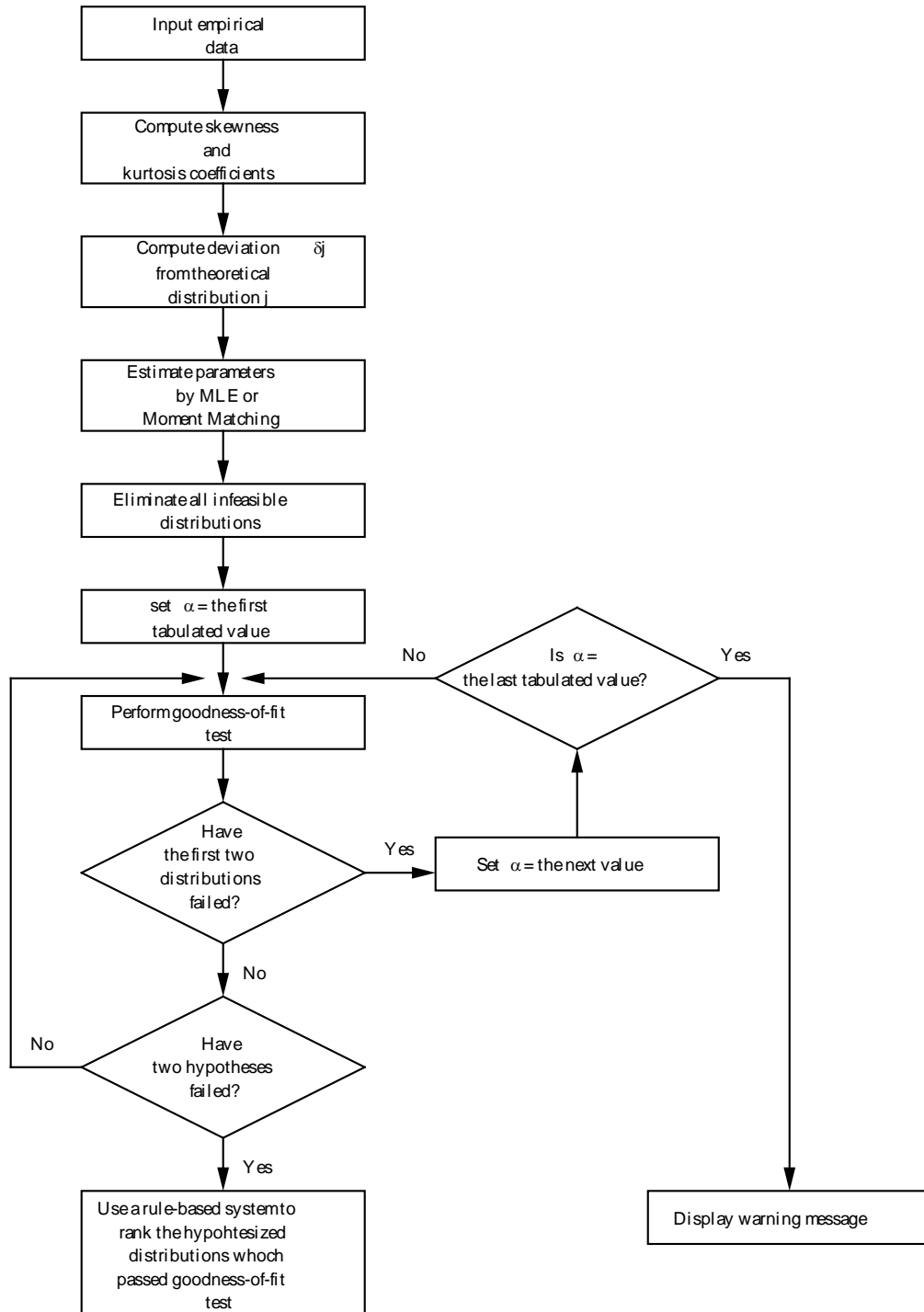


Figure 2.4: Automated Distribution Fitting

3. Rank the distribution using computed δ_j by ascending order. Eliminate any which are not feasible on the basis of data type (continuous vs. discrete) or range of parameter produced. Set $\alpha = \text{first } \alpha (0.1.)$
4. Perform the appropriate goodness-of-fit test on the remaining distributions in the order of δ_j .
5. If the first two distributions are failed, set α to next α (see below), and go to Step 4. If α passes last $\alpha (.01)$, stop and print warning message to user. Otherwise, stop when two hypotheses have finally failed, and go to Step 6.
6. Using a rule-based decision-making system to rank all the distribution which passed at Step 5. The knowledge base may include statistical knowledge as well as ground domain knowledge. (Domain knowledge in a simulation study might be "The time to complete customer service or machine repair might be modeled by a Gamma distribution.") In our current system, the ranking is in ascending order of CPU run-times required to generate samples from the theoretical distributions.

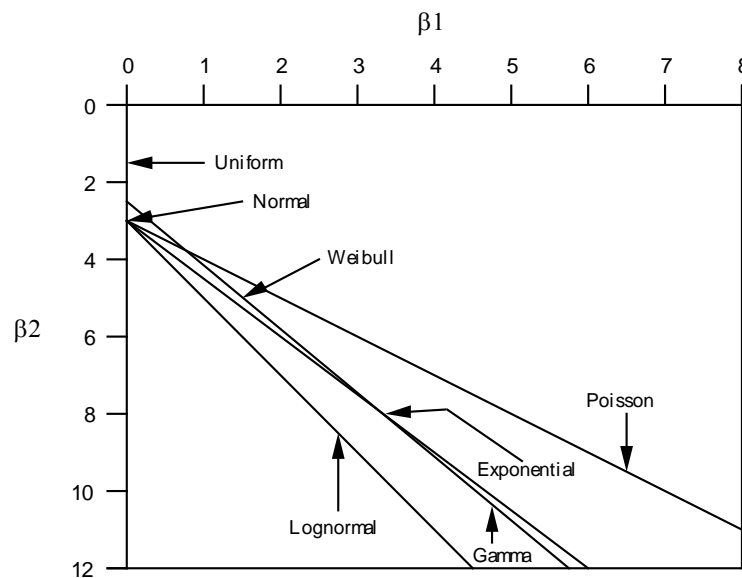


Figure 2.5: Plot of β_1 versus β_2

At Step 4, neither χ^2 nor K-S tests can accommodate all goodness-of-fit tests. It is widely accepted that K-S tests are generally better than χ^2 tests. However, the application areas of K-S tests are limited. For demonstration purposes, the approach adopted in this research is to use the K-S tests. χ^2 testing is employed only when the K-S test is unavailable. Better results can be obtained by applying special tests to a specific hypothesized distribution form.

At Step 5, select α values are used. Those values are the commonly tabulated ones, namely, $\alpha = (.1, .05, .01)$. When distributions do not pass the goodness-of-fit test at $\alpha = .01$, very unusual data is involved. In that case, a message to the user is printed on the screen.

2.6. Fundamental Concepts of Generic/Specific Modeling

Simulation has been proposed as a method whereby manufacturing system designers are able to evaluate the effects of operating scenarios on existing or proposed systems (time in system, inventory levels, resource utilization). Simulation is most often applied by following a process similar to the ten steps outlined by Pritsker (1986.) As defined by Pritsker, these ten steps consist of problem formulation, model building, data acquisition, model translation, verification, validation, strategic and tactical planning, experimentation, analysis of results, and implementation (Mackulak and Cochran 1989).

This section discusses the background behind the development of the generic/specific modeling concept, a concept that simplifies simulation steps, allowing even casual modelers the means to effectively create simulation models. The section gives a brief overview of MASCOT, our first attempt at generic/ specific modeling (Mackulak and Cochran 1988).

Most of the work discussed under the development of Intelligent Front Ends (IFEs) has addressed the issue of model coding (O'Keefe 1985). Currently available commercial packages indicate a popular feeling that simulation would find greater application if modelers did not need to write any code. For a study performed for Motorola, Inc. (Mackulak and Shunk 1986), a contrary discovery occurred. Investigation indicated that the difficulty in the modeling was not the creation of code, but rather the creation of the system abstraction on which the code is based. An incorrect abstraction leads to a plethora of common simulation errors. Some of the more common include over-detailed models, models that don't allow for production of the desired statistics, models for which input data is unobtainable, and models that produce results that contradict statistically correct analysis procedures.

This research therefore concentrates on an alternative approach; the concept of generic-specific models. If a prior model could be created that is flexible enough to fit a wide range of applications, yet specific enough to provide statistically useful results, a

modeler could select this model and avoid the error-prone step of model construction. As a bonus, the model would not need to be coded. The key to implementing this approach is the creation of a widely applicable finite set of generic models.

2.7. IntelliSIM Architecture

Among the current trends in simulation, most of the existing systems have difficulties in modeling the system and in providing an intelligent system environment. The primary problem is that modelers view all simulation models as unique, and tend to recreate topologically similar systems when only the parameters are different. Another problem is that the current systems do not provide user design goals. Providing user design goals is important, because experimental design selection and specific model structure are affected by these goals (Hong, et al. 1989.)

In this section, we present a specification of an architecture for intelligent simulation environments for discrete manufacturing systems. In addition, we present a new methodological approach for the simulation process and compare a traditional simulation process with our approach. Finally, we discuss specification implementation in terms of hardware and software configuration and languages needed. The realization of this architecture is called IntelliSIM.

2.7.1 Methodological Approach

Simulation is an iterative process in which modelers design the model, decide upon a scenario (inputs), run the experiment, analyze the results, decide upon another scenario, run the experiment, and so on. The traditional simulation process is divided into 10 to 12 steps, depending on the researchers, that provide a typical, sound simulation study to guide a model builder.

The traditional simulation process, however, does not aid modelers in deciding upon an appropriate model, nor in determining how to exercise it to find an answer to a given problem. The approach used here combines many of the steps of the traditional simulation process with several knowledge bases and data bases, so that the modelers responsibilities are altered. Our approach is divided into 6 steps: 1) design goal formulation, 2) generic model selection, 3) specific model generation, 4) execution of experiments, 5) output interpretation and validation, and 6) documentation and

implementation. The traditional simulation process is compared with our approach in Table 2.3.

Table 2.3: Comparison of New Approach and Old Approach to Discrete Simulation of Manufacturing Processes

New Approach	Old Approach
<u>Design Goal formulation</u> Formulate user goals GT code to identify specific user design goals through related knowledge base, based on information provided by the user (computer controlled).	<u>Problem Formulation</u> Define the problem to be studied including a statement of the problem-solving objectives (manual effort).
<u>Generic Model Selection</u> Generate GT code for physical system in accordance with the abstraction of the system to serve as a basis to select the particular generic system to be simulated (computer controlled).	<u>Model Building</u> Abstract the system into mathematical-logical relationships in accordance with the problem formulation (manual effort).
<u>Specific Model Generation</u> Instantiate components in the system and then connect them with data provided by permanent factory database. Hands-on data for specific nature might be collected. All the components are provided based on the generic model so that this step depends highly on the type of generic model (computer controlled).	<u>Data Acquisition</u> Identify, specify, and collect data (manual effort). <u>Model Translation</u> Prepare the model for processing (manual effort). <u>Verification and Validation</u> Establish that the computer program executes as intended and that a desired accuracy or correspondence exists between the simulation and the real system (manual effort).
<u>Execution of Experiments</u> Select experimental design through statistics knowledge base which assists in the experimental design selection and prevents duplicating the same simulation runs. After the selection of the experimental design, the executable model will be run to produce the output results (computer controlled).	<u>Strategic and Tactical Planning</u> Establish the experimental conditions for using the model (manual effort). <u>Experimentation</u> Execute the simulation model to obtain output results (computer controlled).
<u>Output Interpretation and Validation</u> Output results are interpreted and validated by the statistics knowledge base which provides several statistical system performance measures and tests (computer controlled.)	<u>Analysis of results</u> Analyze the simulation outputs to draw inferences and make recommendations for problem resolution (manual effort.)
<u>Documentation and Implementation</u> Simulation results are placed in files for generating reports based on simulation activities and results as well as on documenting the model for latter use.	<u>Documentation and Implementation</u> Implement decisions resulting from the simulation and document the model and its use.

2.7.2. Architecture Specification

A user-friendly intelligent simulation environment selects the appropriate generic model from related knowledge bases through a user interactive expert system. The knowledge bases also contain and use the group technology (GT) meta-code for user goals and the GT code for the physical system when selecting a model. User input generates a specific model by providing data, selecting experimental designs, running the executable model, and providing standard outputs. In this way a generic model becomes specific through user interaction. Plots, histograms, confidence intervals of performance measures, and so on are also included when appropriate. This not only allows the user to model an appropriate system, but also decreases a user's fallacies in modeling the system and in analyzing output results.

Figure 2.6 shows a specification of an architecture for an intelligent simulation environment. This environment is divided into a preprocessor and postprocessor environment. The preprocessor environment consists of design goal formulation, generic model selection, and specific model generation steps, and explains how the specific model is built once the problem is defined (see section 2.5.1, "The GT Scheme"). The post-processor environment consists of experimental design, execution of experiments, output interpretation and validation, and documentation and implementation steps. In addition, Table 2.4 shows the contents of the knowledge bases and data bases which were used in these environments.

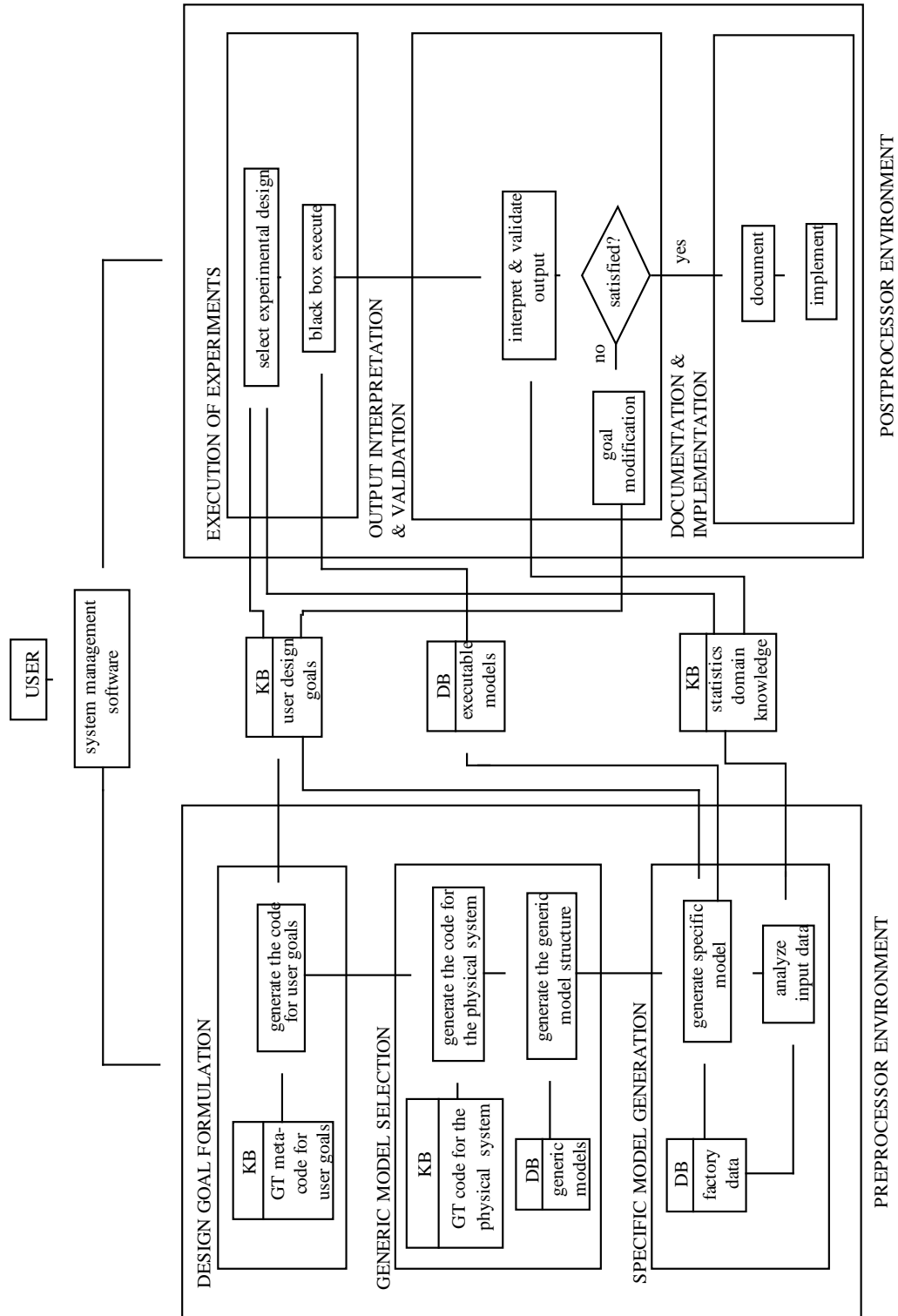


Figure 2.6: Specification of an Architecture for Intelligent Simulation Environments

Table 2.4: Contents of Knowledge Base and Data Base in Intelligent Simulation Environment Architecture

Name of KB or DB	Contents of KB or DB
GT META-CODE FOR USER GOALS (KB)	GT classification and coding scheme for the possible objectives of using simulation in manufacturing system design which will be used to identify specific user design goals.
GT CODE FOR THE PHYSICAL SYSTEM (KB)	GT classification and coding scheme for the physical manufacturing system characteristics which will serve as a basis to select particular generic system to be simulated.
USER DESIGN GOALS (KB)	The specific GT code of user design goals which is generated by interrogating the user and will constitute a basis for specific model generation and experimental design selection.
STATISTICS DOMAIN KNOWLEDGE (KB)	Statistical methods for input and output data analysis as well as validation.
GENERIC MODELS (DB)	Generic models/ submodels from which specific user models will be generated based on the GT codes for the user goals and the physical system.
FACTORY DATA (DB)	Permanent factory database including operation routings, cost data, processing times, reliability data, and resource availabilities from which data required for a simulation run will be extracted.
EXECUTABLE MODELS (DB)	Ready-for-execution models generated by modifying and configuring the generic model structures.

2.7.2.1. Design Goal Formulation

Identifying the user's goals is an essential step in developing a useful simulation experiment. The GT meta-code knowledge base for user goals contains a group technology classification and coding scheme that encompasses most simulation situations. By accessing this knowledge base, the Intelligent Simulation Environment is able to assess the user's needs and identify the simulation goals. Furthermore, user goal

alternatives identified since the user might have different design goals within the same physical system. This step constitutes a basis for specific model generation and experimental design selection.

2.7.2.2. Generic Model Selection

In this step, the GT code for the physical system is generated. The physical system GT code knowledge base provides group technology classification and a coding scheme for the physical manufacturing system characteristics. These characteristics will serve as a basis to select the particular generic system to be simulated. The end goal of this step is to avoid generating variations of previously developed models. After creating a GT code for the user design goals and the physical system, a generic model is selected.

2.7.2.3. Specific Model Generation

Once a generic model is created, the model is completely structured by simply instantiating components (or elements) in the system and then connecting them with data provided by a permanent factory database. This data includes operation routings, processing times, resource availabilities, manufacturing orders, cost data, reliability data, and manufacturing scheduling, from which data required for a simulation run will be extracted.

Input data is analyzed by a statistics domain knowledge base which contains statistical methods for input data analysis such as probability distribution fitting. IntelliSIM provides the user with options for configuring the components in the system based on the selected generic model. This step depends highly on the type of generic model being used. The configured model is then interpreted by the SIMAN simulation program, which produces the executable model. These executable models are stored in an executable models data base, and can be modified later if the user is not satisfied with output results.

2.7.2.4. Execution of Experiments

Before running the executable model, an experimental design is selected based on the user design goals through the statistics domain knowledge base. This knowledge base provides methods for selecting the number of runs, and for the starting conditions of simulation runs. After selecting the experimental design, the executable model is run and output results are collected.

2.7.2.5. Output Interpretation and Validation

Running the executable model generates output results which are interpreted and validated in this step. The user has access to the simulation results as well as SIMAN's output processor. If the user is not satisfied with the results, he may return to the preprocessor environment to modify user goals or the model.

2.7.2.6. Documentation and Implementation

This step contains the report generator which provides knowledge for generating reports in various forms (tabular or graphical) based on simulation activities and results, as well as for documenting the model for later use.

2.8. References

- Adelberger, H.H, "Rule Based Object Oriented Simulation Systems", Proceedings of Intelligent Simulation Environments, 17:1, pp. 107-112 (1986)
- Banks, J and J. S. Carson, Discrete-Event System Simulation, Prentice-Hall, Englewood Cliffs, N. J. (1984)
- Ben-Arieh, D., "Acknowledge Based System for Simulation and Control of a CIM", Proceedings European Conference in Artificial Intelligence and Simulation, July (1986)
- Cochran, J.K. and Cheng, C.S., "Automating the Procedure for Analyzing Univariate Statistics in Computer Simulation Contexts", Transactions of the Society for Computer Simulation (1989)
- Cochran, J.K. and Mackulak, G., "Low Cost Artificial Intelligence Can Aid Design and Simulation", CIM Review, 3:3, pp. 33-36 (1987)
- Cochran, J.K., Mackulak, G.T., Castillo, D., and Du, E., "Configuring Available Software Into an AI/ES Environment for Automated Manufacturing Simulation Design on the PC", Conference on Simulation of Computer Integrated Manufacturing Systems and robotics, San Diego, CA, pp. 1-7 (1987)
- Crowe, Thomas J., Cochran, Jeffery K., and Stahlman, Edward J., "Design Features of a CIM Simulation Planning Tool Based on a C Language Coroutine Implementation Strategy", TIMS/ORSA Joint National Meeting, Las Vegas, NV (1990)
- Henriksen, James O., "The Integrated Simulation Environment (Simulation Software of the 1990's)", Operations Research, 31:6, pp. 1053-1065 (1983)
- Hong, S.C., Cochran, J.K., and Mackulak, G.T., "Specification of an Architecture for Intelligent Simulation Environments", Society for Computer Simulation Series, 20:3, pp. 77-82 (1989)
- Kawamura, K., "Coupling Symbolic and Numerical Computations", IEEE, March, pp. 507-510 (1985)
- Law, A.M., and Kelton, W.D., "Relative Width Sequential Confidence Intervals for the Mean", Communications in Statistics - Simulation and Computation, B10:1, pp. 29-39 (1981)
- Lin, C.Y., Real-Time Automated Output Analysis, Master's Thesis, Arizona State University, Tempe, AZ (1988.)

- Mackulak, G.T. and Cochran, J.K., "Generic/ Specific Modeling: An Improvement to CIM Simulation techniques", International Conference on Modeling and Simulation for Optimization of Manufacturing Systems Design and Application, Tempe, AZ (1989)
- Mackulak, G.T. and Cochran, J.K., "MASCOT: A PROLOG-Based Simulation Modeling and Training Environment", Chapter II.1, Modeling and Simulation Methodology: Knowledge System Paradigms, Elzas, Oren, and Zeigler editors, North Holland, Elsevier Science Publishers (1988)
- Mackulak, G.T., Cochran, J.K., and Shunk, D.L., Generic System Simulation Development on the PC (Phase I-III), Arizona State University, Tempe, AZ (1987)
- Mackulak, G.T. and Shunk, D., Generic System Simulation Development on the PC (Phase 0 - User Needs Definition), Arizona State University, Tempe, AZ (1987)
- O'Keefe, R., "Simulation and Expert Systems - A Taxonomy and Some Examples", Simulation, 46:1, pp. 275-284 (1985)
- Ozdemirel, N.E., Generic Manufacturing Simulation Model Construction Based on group Technology Classification of Models, PhD Dissertation, Arizona State University, Tempe, AZ (1990)
- Ozdemirel, N.E., Mackulak, G.T., and Cochran, J.K., "A Preliminary Group-Technology Classification Scheme for Manufacturing Simulation Models", Winter Simulation Conference, pp. 697-703, San Diego, CA (1988)
- Pegden, C.D., Introduction to SIMAN, Systems Modeling Corporation, State College, Pennsylvania (1987)
- Pettitt, A.N. and Stephens, M.A., "The Kolmogorov-Smirnov Goodness-of-Fit Statistic With Discrete and Group Data", Technometrics, 19, pp. 205-210 (1977)
- Pritsker, A.A.B., Introduction to Simulation and SLAM II, Third Ed., Halsted Press, New York. (1986)
- Reddy, Y.V. Ramana, Mark S. Fox, Nizwer Husain, and Malcolm McRoberts, "The Knowledge-Based Simulation System", IEEE Software, March, pp. 26-37 (1986)
- Schmidt, Bernd, "The Way Ahead in Discrete Simulation, A Critical Assessment of Language and Packages for the Simulation of Discrete Systems," 1980 Summer Simulation Conference Proceedings, Seattle, Washington, pp. 46-49 (1981)
- Schruben, L.W., "Confidence Interval Estimation Using Standardized Time Series", Operations Research, 31:6, pp. 1090-1108 (1983)
- Shannon, Robert E., Richard Mayer, and Heimo H. Adelsberger, "Expert Systems and Simulation", Simulation, 44:6, pp. 275-284 (1985)

CHAPTER 3

INSTALLATION PROCEDURES

It is recommended that you read this entire chapter before attempting to install IntelliSIM.

3.1 Distribution Diskettes

The IntelliSIM software is furnished on four high-density floppy disks (1.2 Mbytes or 1.44 Mbytes). The contents of the disks are as follows:

- **SYSTEM** - contains the IntelliSIM executable programs
- **SEMIAUTO** - contains additional IntelliSIM executable programs and files that are necessary to link IntelliSIM with SIMAN IV.
- **MODELS** - contains the generic models used by IntelliSIM
- **SPECIFIC MODELS** - contains several example models configured by IntelliSIM (i.e., specific models). Each of the ten generic models has a specific model that can be used.

See Appendix A and the READ_ME.DOC on the **SYSTEM** disk for a listing of where the files on the distribution disk are loaded by the installation program.

3.2 Hardware Requirements

- Computer with a 80286/80386/80486 microprocessor (or the equivalent) with a color monitor.
- A 1.2 Mbyte, 5-1/4 inch disk drive **or** 1.44 MByte, 3-1/2 inch disk drive.
- Minimum 7.5 Mbytes available disk space on a hard drive:
 - 4.0 Mbytes for IntelliSIM
 - 3.0 Mbytes for the required SIMAN files
 - 0.5 Mbyte for models and temporary storage.
- Minimum 600K available main memory (this must be 600K of unused memory, not merely installed memory). Keep memory resident programs at a minimum when running IntelliSIM. If not enough memory is available, an error message such as “cannot execute SIMAN.EXE” or “cannot execute AUTO.EXE” will be shown.

3.3. Software Requirements

- DOS 5.0
- SIMAN IV - Version 1.2 (see Section 1.4)
- Library file for Microsoft FORTRAN 5.0 or higher (see Section 3.5)
- Library file for Microsoft C 5.1 or higher (see Section 3.5)
- To use IntelliSIM, SIMAN must be included in the PATH statement.

3.4. SIMAN Requirement

The user needs to provide, at his own expense, SIMAN IV - Version 1.2 for execution of the simulation models generated by IntelliSIM. SIMAN is not furnished as part of the IntelliSIM installation software. If the user does not have the SIMAN package, he can still use IntelliSIM to select and configure a generic model, but will not be able to run the specific models. The SIMAN simulation language can be purchased from:

System Modeling Corporation
504 Beaver Street
Sewickley, Pennsylvania 15143
(412) 741-3727
(412) 741-5635 (Fax)

3.5. Library File Requirement

It is necessary to link several IntelliSIM modules with SIMAN and MS-FORTRAN (5.0 or higher) and MS-C (5.1 or higher) library files. The installation program will prompt for the library files names (including the full path name) to link IntelliSIM and SIMAN.

Because SIMAN uses the large model library, the FORTRAN and C library files should also be large model. For more information about library options, please read the SIMAN IV manual titled **System Operations and Installation Guide** (pages 3 - 4).

3.6. Installing IntelliSIM

To install IntelliSIM, insert the disk labeled **SYSTEM** into drive A or B and type the following:

```
C:\> A: <RETURN>          C:\> B: <RETURN>
A:\> INSTALL <RETURN>      or  B:\> INSTALL <RETURN>
```

Follow the on-screen instruction for installing IntelliSIM. Possible error messages which may appear are:

- **Not Enough Space** - less than 4.0 MBytes of disk space is available on the hard drive.
- **Incorrect DOS Version** - the version of DOS is lower than DOS 3.3.

If no errors occur, IntelliSIM installs under the specified directory (specified by the user in the installation program). The default directory is ISIM.

After all necessary IntelliSIM files have been transferred to the specified directory, the next step of the installation procedure is to link the IntelliSIM modules with SIMAN and the library files. The user is required to enter:

- The path statement for the SIMAN directory. For examples, C:\SIMAN.
- The path statement for the MS-FORTRAN library file, For example, C:\LIBRARY\FORTRAN\LLIBFORE.LIB.
- The path statement for the MS-C library file, For example, C:\LIBRARY\C\LLIBCE.LIB.

The input procedure “loops” until all of the above fields have been entered. Once this is completed, the installation program will perform the linking and the program will be ready for use.

If the user does not have SIMAN or the necessary library files, the linking process can be aborted by pressing the <ESC> key. The linking step can be completed at a later time by running the program LINKRSP.EXE. To link with LINKRSP.EXE, go to the IntelliSIM directory (default ISIM) and type: **LINKRSP**. Follow the on-screen instructions and the linking will be performed.

CHAPTER 4

GETTING STARTED WITH INTELLISIM

This chapter provides a quick overview of how to start and run IntelliSIM. The chapter concludes with an in-depth example on how to create, run, and analyze a simulation project with the IntelliSIM environment.

modify picture/ add screen printouts

this page left blank for menu diagram (FIG 4.1)

Table 4.1. IntelliSIM Menu Options

Menu number/ Menu options (M - Menu, A - Action)	Description
M0 Main menu	
A1	Tutorial
M2	Select a model
M3	Configure selected model
M4	Execute Simulation
A5	Log off
M2	Select a model
M21	Select among the generic models
M22	Select among the specific models
A23	Consult the expert system
M21	List of generic models available
M22	List of specific models available
M3 Model configuration	
M32	Work stations
M33	Work-in-process storage areas
M34	Optional and special resources
M35	Material handling system
M36	Jobs
M37	Operation routings
M38	Equipment breakdown/maintenance
M39	Fit data distribution
M310	Output statistics
A311	Model status summary
A312	Exit model configuration
M32	Work stations
A321	Add a new work station
M322	Update existing work stations
M322	Update existing work stations
A3221	Station types
A3222	Primary resource capacities
A3223	1st optional resource assignments
A3224	2nd optional resource assignments
M3225	Processing time distributions
A3226	Alternatives for machining stations
A3227	Failure rates for inspection stations
M3225	Processing time distributions
A32251	Setup time distributions
A32252	Processing time distributions
A32253	Loading/unloading time distributions
A32254	Efficiency ratios

Table 4.1. cont.

Menu number/ Menu options (M - Menu, A - Action)	Description
M33	Work-in-process storage areas
A331	Input buffer capacities
A332	Output buffer capacities
M333	Balking options
M334	Priority rule
M333	Balking options
A3331	Wait on transportation media
A3332	Block previous station
A3333	Transfer to common storage area
M334	List of priority rules available
A3341	FIFO - first come first served
A3342	User assigned priorities
A3343	SPT - shortest process time
A3344	LPT - longest process time
A3345	SRPT - shortest remaining process time
A3346	LRPT - longest remaining process time
A3347	EDD - Earliest due date
A3348	Minimum slack
M34	Optional and special resources
A341	1st set of optional resources
A342	2nd set of optional resources
A343	Pallets
A344	Fixtures
M35	Material handling system
A351	Manual transportation
M352	Transporters
M353	Conveyors
M352	Transporters
A3521	Transporter type 1
A3522	Transporter type 2
A3523	Distances for transporters
M353	Conveyors
A3531	Conveyor type 1
A3532	Conveyor type 2
A3533	Segments for conveyor type 1
A3534	Segments for conveyor type 2
M36	Jobs
A361	Add a new job
A362	Update existing jobs

Table 4.1. cont.

Menu number/ Menu options (M - Menu, A - Action)	Description
M362	Update an existing job
M3621	Job arrivals
A3622	Pallet types
A3623	Fixture types
A3624	Batch sizes
A3625	User assigned priorities
A3626	Due dates
A3627	Match numbers for assembly
M3621	Job arrivals
A36211	Job arrival distribution type or number of batches per job
A36212	Percentage job types or job arrival parameters or arrival times for batches
M37	Jobs Routings
A371	Define a new jobs routing
A372	Add operations to a routing
A373	Delete operations from a routing
M374	Change operations in a routing
A375	Material handling equipment types
M374	Change operations
A3741	Change station numbers
A3742	Change setup times
A3743	Change processing times
A3744	Change loading/unloading times
M38	Equipment breakdown/maintenance
A381	Time between failures distribution type
A382	Time to finish repair distribution type
A383	Time between maintenance distribution type
A384	Time to finish maintenance distribution type
A385	Specify breakdowns for stations
A386	Specify maintenance for stations
M39	Fit data distribution
M391	Specify data and write to a file
M392	Execute DISTFIT with a data file
M310	Output statistics
A3101	Statistics for work stations
A3102	Statistics for optional resources

Table 4.1. cont.

Menu number/ Menu options (M - Menu, A - Action)	Description
M4Execute Simulation	
M41	Manual execution of specific model
M42	Semi-automatic experimental design
M43	View and Edit Summary Reports
A44	Goto Main Menu
M41	Manual execution of specific model
A411	Number of replications
A412	Beginning time of each batch
A413	Ending time of each batch
M42	Semi-automatic experimental design
A421	Select a specific model
A422	Priority variable selection
M43	View and Edit Summary Report
A431	View Summary Report
A432	Edit Summary report
A433	Print Summary Report
A434	Previous Menu

4.1. Executing IntelliSIM

After completing the installation procedure, IntelliSIM can be executed by changing the directory to the IntelliSIM directory and then typing ISIM. For example:

```
CD C:\ISIM <CR>
ISIM <CR>
```

The first screen the user will see when executing IntelliSIM is the introductory screen. Hit any key to continue. The main menu will appear. The main menu options provided by IntelliSIM for model selection, configuration, and Execute Simulation are described in Chapters 4, 5, and 6.

4.2. Using the Menus

To exit any IntelliSIM menu the user must press the <Esc> key or select the menu exit option. The only exception to this is when IntelliSIM is expecting a numeric input

rather than the choice of a menu option. In that case the <Esc> key will not work, and some numeric value must be entered. The user should check the status line (the last line on the screen with reverse video) as to what value he should enter to exit from a certain option.

The hierarchical menu structure of IntelliSIM is shown in Figure 4.1. Menu options are described in Table 4.1. The menu options in Figure 4.1 and Table 4.1 are named in accordance with the hierarchical menu structure. The first letter used in naming indicates the type of action taken when an option is selected. An “M” indicates that another menu appears when that option is selected. An “A” implies that an action is taken to configure the model by the user, or the system, or both as in most cases. At the end of each action, the data entered or the parameters changed by the user are recorded in the specific model file.

Model selection can be done by selecting a generic model by name, selecting a specific model from a floppy disk (in drive A), or selecting a generic model by consulting the IntelliSIM expert system. To select among the specific models, the user must have specific model(s) previously generated and stored on a floppy disk. Once a model is selected, it can be configured using the model configuration options. When model configuration is complete, a new specific model is created which may be executed.

4.3. Saving Specific Models

If the user wants to select one of the generic models, configure the generic model, and save the specific model generated permanently, a formatted floppy disk must be available. IntelliSIM will only save models on the A-drive and will not allow the user to format a floppy disk once a session has started.

4.4. Helpful Hints

Since IntelliSIM **does not** confirm that specific models are fully specified, it is possible that a user may try to execute a model with data inconsistencies. This will lead to SIMAN run time errors, inaccurate models, and erroneous results. These hints are designed to point out areas where errors are likely to occur and how to avoid them.

- When first using IntelliSIM, it is best to explore one or more of the examples from the SPECIFIC MODELS disk, rather than attempting to build a new specific model. This will provide a feeling of how the system works and allow a model to be executed immediately. The CONFIGURE SELECTED MODEL option of the main menu can be used to explore these models. For each of the ten generic models, there is an example configured model on the diskette.
- The first step to building a new model is to select a generic model. By default a newly selected model is empty and needs to be configured to add details about the system such as arrival rates, job types, and routings. The model may be run without any configuration but will result in all statistics being equal to 0.0. The reason for this is that no routing is specified for parts. They are disposed as soon as they are created, without any operations being performed on them.
- The first modifications that should be done to a new model is to specify a job routing for job type 1. Initially there is only one job type, but if more job types are added, be sure to specify routings for them; they will not be added by default. Failure to specify job routings for new jobs will result in SIMAN run time errors.
- When configuring a specific model it is often necessary to modify several parts of the generic model to fully incorporate a single function. For example, if a new part is added to the model, a routing must also be specified for the new job type. If it is not, a SIMAN run time error will occur.
- When using an assemble operation, two characteristics must be added. First, there must be at least two job types. If a second job type is not added to the model, no

assembly can occur since the assembly operation requires at least two different job types. If the system only has one job type the assembly load queue will continue to fill until the simulation is terminated or a SIMAN run time error occurs due to too many entities in the system. Look in the workstation menu option of the model configuration menu to see what the default workstations are. Second, a match attribute must be specified so that the assembly operation knows how to match jobs.

4.5. Tutorial/Example of IntelliSIM Session

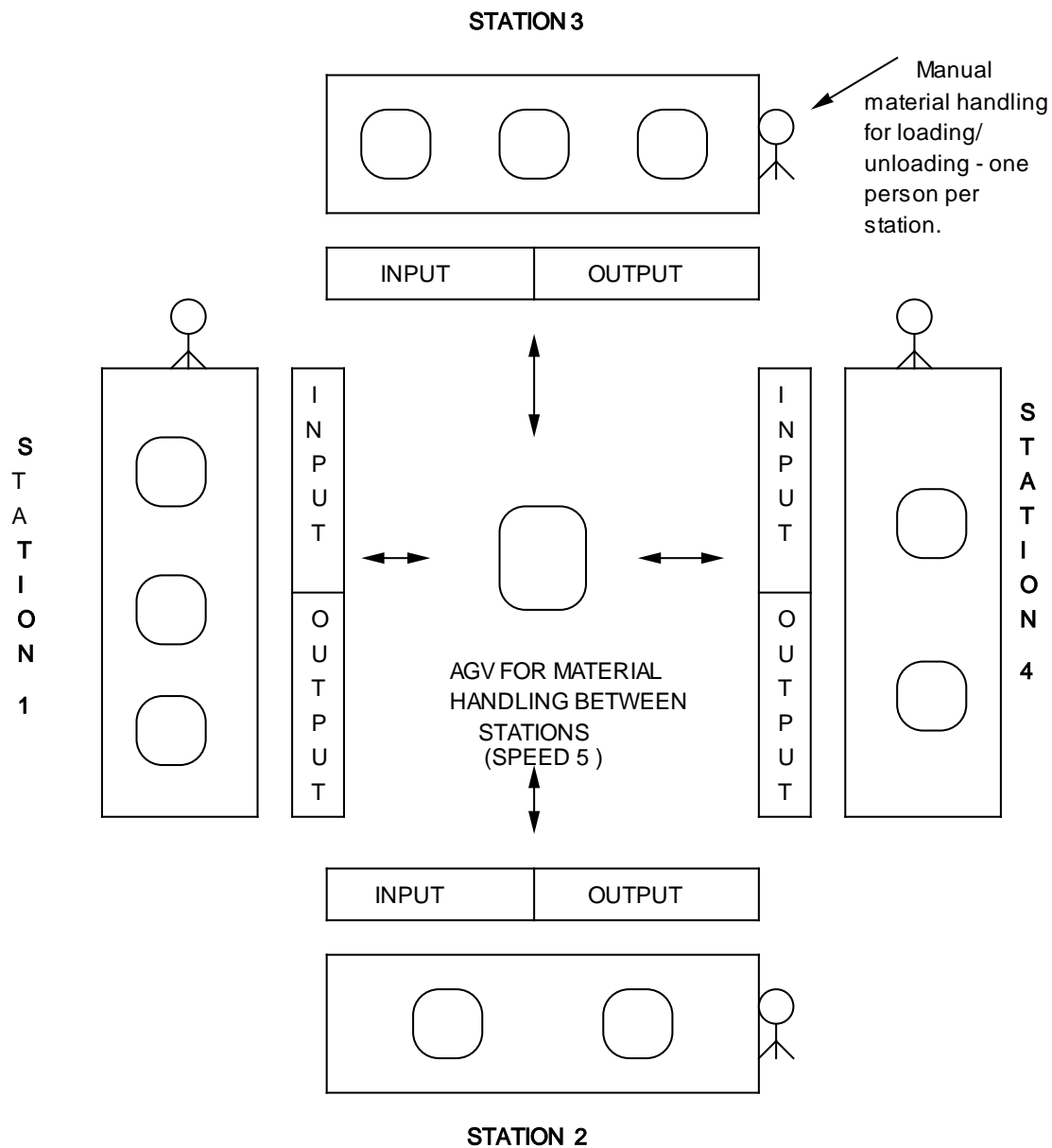
The following section describes in detail how IntelliSIM can be used for solving a particular problem. We have selected this simulation problem in such a way so as to cover as many features of IntelliSIM as possible and to give the user an idea of how different features can be used. However, it should be realized that this is only an illustration and is meant for providing the user with a better understanding of IntelliSIM. IntelliSIM can be used for a vast variety of simulation models apart from that explained in this example.

Description of the system

The production system (see Figure 4.2) consists of three unique manufacturing stations and one packing station. Stations 1 and 3 can run up to 3 machines while stations 2 and 4 each can run up to 2 machines. The distances between all machining and packing station are such that it takes an equal amount of time to move the semi-finished product from one station to another using an AGV. Only one AGV is used for all the movements. Distance between any two stations is 3 distance units, while speed of the AGV is 5 distance units/time unit.

Two products are produced by this manufacturing cell. The raw material that appears at the cell has an underlying exponential distribution with mean of 3.5 time units.

The type of product is randomly determined when a product enters the system. About 60% of the products are Type 1 and 40% are Type 2. There are input and output queue facilities at each of the stations. The capacity of input and output queues at each station is 50 and jobs are forced to wait on the transportation device when the queues are at full capacity. Processing, loading and unloading times for each type of machine is shown in Table 4.2. Loading and unloading operations are manual and there is 4 individuals in the cell performing these operations. Each person is dedicated to one station for loading and unloading purposes. The setup time for the machines is negligible and not considered for the purposes of analysis.

**MAIN DETAILS:**

- LOADING AND UNLOADING MANUALLY - SECONDARY RESOURCE - CAPACITY 4 ONE PERSON PER STATION.
- THREE PARALLEL IDENTICAL MACHINES FOR MACHINING STATIONS 1 & 3
- TWO PARALLEL IDENTICAL MACHINES FOR STATIONS 2 & 4
- SEQUENCES: JOB TYPE 1 : 1-2-3-4 : JOB TYPE 2 : 1-3-4

Figure 4.2. Physical Layout of the Manufacturing Cell

Table 4.2. Processing, Loading and Unloading Time for Each of the Stations in the Manufacturing Cell.

STATION	PROCESSING TIME	LOAD/UNLOAD TIME
1	uniform(2.0,3.0)	uniform(1.0,1.5)
2	uniform(3.0,4.0)	uniform(1.0,2.0)
3	uniform(2.0,3.0)	uniform(1.0,1.5)
4	uniform(2.0,3.0)	uniform(1.0,1.5)

The production sequence for each product is the following:

- Type 1: station 1 - station 2 - station 3 - packing station
- Type 2: station 1 - station 3 - packing station.

The primary purpose of this simulation is to study utilization of resources and remove bottlenecks if any by increasing number of resources. Initially, the simulation run length is 500 time units.

STEP BY STEP DESCRIPTION

This section is a step-by-step description of the simulation process using IntelliSIM. It is designed to provide the user with an understanding of how IntelliSIM works.

STEP 1

To start the IntelliSIM program, type ISIM in the appropriate directory. For example,

CD C:\ISIM <CR> (change to the IntelliSIM directory)

ISIM <CR> (start the program)

STEP 2

After the initial screen, IntelliSIM displays the main program menu. Since we are generating/creating a new model, select **Select a Model** (see Figure 4.3. and 4.4).

Figure 4.3. Main Menu

Figure 4.4. Select a Model

STEP 3

A new menu is displayed. Choose **Select Among Generic Models** for we are generating a new model and not utilizing a previously generated specific model. If the user is not familiar with the different generic models, select the **Consult the Expert System** option. The Expert system will assist the user in the selection of an appropriate generic model. For more details about this option, please refer to the Section 4.3. “**Generic Model Selection by Consulting the Expert System**”.

STEP 4

From the list of generic models on the screen, select **GQ1S02.mod**, for our example has one secondary resource which is used for loading and unloading, and there are separate WIP storage areas for both incoming and outgoing jobs. We will select the first optional resource used for the entire operation, even though our model does not have an optional resource which is used for the entire operation. IntelliSIM has the capability to later exclude this optional resource from the actual simulation of the model. After the selection, IntelliSIM verifies the selected model by asking “Do you want to select this model ?” Answer **yes**.

STEP 5

From the next menu, select **A1 - Job types are specified as percentages of random arrivals**, since in our model 60% of the total randomly arriving jobs are Type 1 while the other 40% are of Type 2.

STEP 6

This completes the selection of a model. The next step is to configure (make specific) the selected model by setting various parameters and variables values for the model. Select the **Configure Selected Model** option (see Figure 4.5).

Figure 4.5. Configure Selected Model

STEP 7

IntelliSIM gives a status summary. Press the <Enter> key. This status summary corresponds to the default values assigned to the model. The configuration process will update these values.

STEP 8

From Model Configuration menu, select the **Work Stations** option. From this new menu, select **Update Existing Work Stations** since by default, IntelliSIM defines 6 work stations. For our example, we have only 4 stations, thus there is no need to use **Add a New Work Station** option. From the following menu, select the **Station Types** option. For stations 1 to 4, type **machining** since all four stations are used for part processing. From the same menu, select **Primary Resource Capacities**. For stations 1 and 3, enter capacity 3 and for stations 2 and 4, enter capacity 2. Do not worry about station 5 (which is not relevant to our example) we are going to exclude it by not having it in the routing for any product. Again, from the same menu (Update Existing Work Stations), select **2nd Optional Resource Assignments** for this resource is used for loading and unloading. Note that we are not using the 1st optional resource at all in our model. For

stations 1 to 4, enter 51 to 54 for 2nd optional resource since we have one person dedicated to each station for loading and unloading purposes. Under the Update Existing Work Stations menu, remaining options are Processing Time Distributions, Alternatives for Machining Stations and Failure Rates for Inspection Stations. Due to the following reasons, we are not concerned with any of these:

- 1) Will use default uniform distributions for processing and loading/unloading.

If using any other distribution were needed, go under this menu and select the desired distribution.

- 2) There is no alternative for machining stations in our model.

- 3) There is no Inspection station in our model. If any of these features are used, appropriate information should be provided. Now go to Model Configuration menu by pressing the <Esc> key.

STEP 9

Next select **WIP Storage Areas**. For options **Input Buffer Capacities and Output Buffer Capacities**, enter capacities 50 for each of the 4 stations. For **Balking Options** and **Priority Rule**, enter Wait on Transportation Media and FIFO as it is our model's requirements. Go back to the Model Configuration menu.

STEP 10

Select the **Optional and Special Resources** option. Under this menu, the only option we will need is 2nd Set of optional resources since we are not concerned with 1st set of optional resources, pallets, and/or fixtures. Select option **2nd Set of Optional Resources**. Enter the capacity of 51, 52, 53 and 54 optional resource as 1 (actually, this capacity is default). Go back to Model Configuration menu. If the 1st set of optional resources or pallets or fixtures are used in a model, then all the details should be provided by selecting the appropriate options.

STEP 11

Select now the **Material Handling System** option. From the following menu, select the **Transporters** option for our system using an AGV for material handling functions. Select **Transporter Type 1** under Transporters menu. Enter the capacity as 1 and velocity as 5. Now select **Distances for Transporters** under Transporter menu. Enter distances from station 1 to stations 2, 3 and 4 as 3 units. The option Transporter Type 2 is not used for our example only utilizes a single transporter for material handling. Go back to the Model Configuration menu.

STEP 12

Select the **Jobs** option. Select **Add a New Job Type**. By selecting this option, the next job number is assigned to the model (as a default, IntelliSIM has a job type 1 already defined). In our example, we have 2 job types and thus have to add job type 2 (job type 1 is defined by default). If more than two job types are needed, select the same option that many times. Next, select the option **Update Existing Jobs** option under Jobs menu. The only option we will use for our model is the **Job Arrivals** option. Select that option. Select the **Job Arrival Distribution** option. As the distribution type is Exponential, select Exponential and enter the mean as 3.5. Under the Job Arrivals menu, select **Cumulative Percentage Job Types** and enter the cumulative percentage of 60 for part type 1 and 100 for part type 2. Go back to the Model Selection Menu. In this example model, many features of IntelliSIM are not used. If used in some other model, the user should provide complete details by selecting appropriate options under the menus.

STEP 13

Select **Jobs Routings** option. Under that menu, select option **Define New Jobs Routings**. For product 1 enter operations routing as station 1 - station 2 - station 3 - station 4. For product 2 enter routing as station 1 - station 3 - station 4. Please note that for each product routing, **Define New Jobs Routings** should be selected again. Also provide the values of appropriate parameters for the distributions used for setup, processing and loading. In our example, provide minimum and maximum values for each

uniform distributions from Table 4.2. Because we are considering setup time to be negligible in this model, assign 0 for both minimum and maximum values for setup time. From the Operations routing menu, the next option to select is the **Material Handling Equipment Types** option since we do not want to add, delete or change operations in a routing. This option can be used in later stages for simulation analysis. In the Material Handling Equipment Types option, enter Transporter type 1 for both job type 1 and 2 for all the stations as this is the material handling equipment type that we have selected previously. Go back to Model Configuration menu.

STEP 14

Note that we are not using any equipment breakdown/failure or fit data distribution. For more details about these options, refer to Chapter 5 in this manual. Select the **Output Statistics** option. From this menu, select “yes” in the collect statistics question, for Work Stations 1 to 4 and optional resources 51 to 54. Go back to the Model Configuration menu.

STEP 15

Select the **Model Status Summary** option. This gives you a brief summary of the parameters and options selected for the model. Check if anything is incorrect and if so, go back to the appropriate option and make corrections.

STEP 16

The model configuration stage is complete, so select **Exit Model Configuration** (see Figure 4.6). Save the file. IntelliSIM saves the file on the “a” drive.

Figure 4.6. Exit Model Configuration

STEP 17

From the main menu, select **Execute Simulation** (see Figure 4.7). Under Execute Simulation, select **Manual Execution of Specific Model**. IntelliSIM will ask for a selection of particular specific model. After a particular model is selected, IntelliSIM asks the user for number of batches, the beginning and ending time of each batch. After this information is given, IntelliSIM runs the selected model. For our model, the output generated by the SIMAN simulation is given on the next page. Of course, a better method is to use Semi-Automatic Experimental Design which is described in the following step.

Figure 4.7. Execute Simulation

STEP 18

One of the most powerful features in the IntelliSIM is Semi-Automatic Analysis. Basically, this feature relieves the user from the task of doing statistical analysis and finding confidence interval. With this feature the user specifies a priority variable. IntelliSIM provides the confidence interval and percentage error in real time. At any point, if the percentage error is acceptable then user can terminate the semiautomatic analysis by pressing the END key. If the user does not want to terminate, IntelliSIM terminates the semiautomatic analysis when percentage error reaches below 5%. For more discussion of Semi-Automatic Experimental Design, refer to Chapter 7 in the user's manual.

For this example, the priority variable is **overall time in the system**. From the Main menu, select **Execute Simulation**. From the Execute Simulation menu, select **Semi-**

Automatic Analysis (see Figure 4.8). After this option is selected, IntelliSIM asks for the selection of particular model. IntelliSIM will ask you to insert the your specific model disk in the “A” drive (the disk on which you saved the model you previously configured). Select the name of model which was used when saving the model. After model is selected, IntelliSIM provides the user with all the output variables available and in this case, user should select the desired priority variable - overall time in the system.

Figure 4.8. Select Semi-Automatic Analysis Option

IntelliSIM will run the model and display the confidence interval and percentage error at the end of each batch. At any point, user can terminate the semiautomatic analysis by pressing the END key. If semiautomatic analysis is not terminated till percentage error reaches below five percent, then IntelliSIM automatically terminates the semiautomatic analysis. At the end, a graphical summary is provided to the user. A user can again view and edit the simulation results by selecting the appropriate options.

Figure 4.9. Confidence Intervals and Percentage Error for Each Batch

Figure 4.10. Graphical Summary of Semi-Automatic Analysis

Figure 4.11. Summary Report

CHAPTER 5

MODEL SELECTION

This section will walk the user through various menu options that are provided in IntelliSIM to select a model. After completing the installation procedure, IntelliSIM can be executed by first changing the directory to the IntelliSIM directory and then typing ISIM. For example:

```
CD C:\ISIM <CR>
ISIM <CR>
```

An introductory screen (Figure 5.1) will appear when the user initiates a IntelliSIM session. The main menu (Figure 5.2) will appear after the user presses any key to clear the introductory screen. On any IntelliSIM menu, one option is highlighted. The user may use the upper or lower arrow keys to switch from one option to another. The highlighted option will be selected when the user presses the <Enter> key. The first option in the main menu helps the user to select a model. The user needs to select a model first in order to configure a generic model or save a specific model. When he chooses the **Select a Model** option from the main menu, the model selection menu appears.

As seen in the model selection menu (Figure 5.3), the model selection module of IntelliSIM provides three options for selecting a model: (1) select among the generic model names, (2) select among the previously generated specific model names, and (3) consult the expert system. The first two options are provided for the modelers who have used the model building system before and are familiar with the generic models. The inexperienced user should consult the expert system for model selection.

Figure 5.1. First Introductory Screen of IntelliSIM

Figure 5.2. Main Menu

Figure 5.3. Model Selection Menu

5.1. Generic Model Selection by Name

When the first option, **Select Among the Generic Model Names**, is selected from the model selection menu, a menu of the available generic model names is displayed (Figure 5.4). There are ten generic models in the model base of IntelliSIM, although combinations of four job arrival patterns with these ten models allow for forty unique generic models. The generic model menus are based on the Group Technology (GT) code. The user may refer to Section 1.4 of this manual for a more complete discussion of the GT classification and coding scheme, and for an explanation of key GT digits. The key digit A* for the job arrival pattern is not included in the names on this menu, reducing the number of model names listed. Because these names differ by arrival pattern only, it would be needlessly complex for the user to remember what each name means. Instead, by using the main model types, the user differentiates among ten models. An additional menu is provided for selection of the job arrival pattern.

When the user attempts to choose one of the model names by pressing the <Enter> key when the name is highlighted, the properties of that model that are defined by key GT digits Q* and S** will be displayed, the user will be asked if he wants to select the model (Figure 5.5). If the user decides to select the model, a menu of the four job arrival patterns will be displayed (Figure 5.6). Model selection will be concluded by choosing one of the arrival patterns.

Figure 5.4. Generic Model Names Menu

Figure 5.5. Description of a Generic Model

Figure 5.6. Job Arrival Patterns Menu for Generic Model Selection by Name

5.2. Specific Model Selection

A modeler who has used the model building system before and generated one or more specific models may choose the second option in the main model selection menu. This option displays the names of the specific models that are stored on a floppy disk and allows the user to select from the list. By definition, only generic models are stored on the hard disk, thereby requiring all specific model storage and maintenance to be directed by the user and subsequently stored on floppy.

Whenever a specific model is saved under a name given by the user, IntelliSIM appends the file extension .FAC to the file name. When he wants to select among the previously created specific models, IntelliSIM will prompt the user to insert a floppy disk on which specific models are stored (Figure 5.7). It will search the directory of the disk for the file names with an extension of .FAC and list those file names in a menu (Figure 5.8).

Figure 5.7. IntelliSIM Prompt for Specific Models Disk

Figure 5.8. Specific Models Menu

5.3. Generic Model Selection by Consulting the Expert system

The third option for model selection, consult the expert system, assists an inexperienced modeler in selecting a generic model. The introductory screen for the IntelliSIM expert system is shown in Figure 5.9. A generic model is selected through three phases which are described below.

5.3.1. Meta-Level Query

The initial questioning phase to decide whether IntelliSIM is capable of handling the user's problem is performed under the meta-level query. The introductory screen for the meta-level query is given in Figure 5.10. The meta-level query is optional. The experienced user may proceed to model selection without having to answer meta-level questions every time he uses the system. Questions that are asked are listed in Figures 5.11. Question responses at this stage compare the scope and general capabilities of the model building system versus user's requirements and the nature of his manufacturing process. The user may answer each question as yes or no, or require an explanation as to why a certain question is asked (Figure 5.12).

The expert system takes certain actions after each question is answered. The answer may be satisfactory, i.e., user's requirements and nature of his process match the system capabilities. In this case, the meta-level query continues without interruption. If all the answers are satisfactory then the expert system draws a successful conclusion as in Figure 5.13. When the answer is unsatisfactory, the user's requirements or process characteristics may be in total disagreement with the system's scope and capabilities, in which case the expert system concludes that IntelliSIM cannot be used to study the user's problem. This conclusion is given in Figure 5.14. If a disagreement is determined non-fatal then the system displays its conclusion, and the user decides if he wants to continue.

Figure 5.9. Introductory Screen for IntelliSIM Expert System

Figure 5.10. Introductory Screen for Meta-Level Query

Figure 5.11. Meta-Level Query Questions

Figure 5.12. Meta-Level Query Explanation Mechanism

Figure 5.13. Successful Conclusion of Meta-Level Query

Figure 5.14. Unsuccessful Conclusion of Meta-Level Query

5.3.2. Assumption Review and Measuring the User Acceptance

Upon the successful conclusion of the meta-level query, the user may choose to review the assumptions of the generic models. The introductory screen for assumption review is shown in Figure 5.15. The assumptions are grouped for ease of understanding, and these groups are provided in a menu (Figure 5.16). The user may choose to review any group of assumptions, and is asked if he accepts or rejects each assumption (Figure 5.17). The percentage of assumptions that are accepted is computed for each assumption group and displayed in the menu as the acceptance score.

Assumptions reflect the limitations of the generic models as well as their strengths. Different levels of all the GT digits that are implemented in the generic models are listed as assumptions, as well as those that are ignored in the implementation. By reviewing the assumptions, the user will be aware of the modeling options that are provided by the generic models and their limitations, and state his opinion on the sufficiency and acceptability of these options. Hence the acceptance scores provide a measure of the user's acceptance of the system.

5.3.3. GT Code Generation and Model Selection

Review of assumptions is followed by the generic model selection. The introductory screen for model selection is shown in Figure 5.18. The expert system assists the user in this selection through two menus given in Figures 5.19, 5.20. The options chosen by the user from these menus determine the key GT digits Q^* , A^* , and S^{**} which are mentioned in Section 1.4, and result in selection of a generic model. The name of the selected model is displayed to the user (Figure 5.21).

Figure 5.15. Introductory Screen for Assumption Review

Figure 5.16. Assumption Review Menu

Figure 5.17. Sample Screen for Assumption Review Menu

Figure 5.18. Introductory Screen for Model Selection Phase

Figure 5.19. First Model Selection Menu

Figure 5.20. Second Model Selection Menu

Figure 5.21. Conclusion of the Model Selection Phase

The GT code for physical system characteristics is partially generated. Only the key digits that are necessary to select a generic model are determined. The rest of the GT code will be defined as the generic model is configured, because the remaining digits are determined as a function of modeling data specifics. These digits result when the user defines his system, and are stored in the SIMAN experimental frame. This feature of the model building system causes a significant portion of the GT code to be implicit, because the digits are hidden in the data specification. The system, however, gains flexibility from the same feature. If the entire GT code were to be defined strictly during the model selection process, the user would not be able to change certain modeling options once the generic model is configured. This flexibility allows modeling options, except for those that are defined with the key digits, to be changed even after a specific model is generated. The user may evaluate different options for his manufacturing process by reconfiguring the same specific model with minimal change, rather than choosing a new generic model and configuring it all over again. Minimization of the configuration requirement is an important feature in IntelliSIM, because the configuration process may become tedious if the number of work stations and number of job types are fairly large.

At the end of the model selection phase the selected generic model is known to the model building system and ready for configuration.

CHAPTER 6

MODEL CONFIGURATION

The model configuration module of IntelliSIM configures the selected generic or specific model and saves the final form of the specific model. The user must select a generic or a specific model prior to choosing the model configuration option. Model configuration involves specification of the data required for all the experimental frame elements to construct a specific model from a generic model. Configuration is carried out through various menus and windows.

6.1. Configuration

When the user chooses the model configuration option from the main menu, the model status summary appears on the screen (Figure 6.1). If the currently selected model is generic, this screen provides a summary of the default configuration of the generic model. For a specific model, the latest configuration is displayed as it was saved by the user. The summary screen allows the user for a quick review of some of the specifics of his model. When the user presses any key the summary screen disappears and the main model configuration menu appears (Figure 6.2). Each of the options in this menu is discussed in the following sections.

Figure 6.1. Model Status Summary Screen

Figure 6.2. Model Configuration Menu

6.2. Work Stations

The work stations option permits the user specification of various characteristics of work stations which are the basic job processing units in generic models. Upon the selection of this option from the configuration menu, the work stations menu appears (Figure 6.4). There are three options in this menu for addition, update, and deletion of work stations.

6.2.1. Addition of a New Work Station

The first option in the work stations menu allows addition of a new work station and specification of the station type (Figure 6.5). There are five different station types in IntelliSIM. The user may read the description of station types under the assumption review phase of the generic model selection by consulting the expert system option (see Section 4.3). Addition of a work station is complete when station type is chosen for the new station.

Figure 6.4. Work Stations Menu

Figure 6.5. Addition of a New Work Station

By default, there are six stations already added in a generic model. These are: (1) machining, (2) assembly, (3) inspection, (4) fixture loading/unloading, (5) batch size independent process, (6) common storage. Among the existing stations, the last station is always the common storage. The maximum number of work stations allowed in any generic model is fifty, including common storage. The default properties for a newly added work station are as follows.

- Station type: User-defined (default machining).
- Primary resource capacity: One.
- First and second optional resource assignments: None.
- Processing time distributions (setup, process, loading/unloading): Uniform.
- Efficiency ratio for processing times: 100 percent.
- Alternative for a machining station: None.
- Failure rate for an inspection station: Zero.
- Input buffer capacity: 9999.
- Output buffer capacity: 9999.

6.2.2. Update of Existing Work Stations

The second option in the work stations menu is concerned with update of the properties of existing work stations. In general, upon addition of a new station, the user needs to select this option to define its properties. The work stations update menu appears when this option is selected (Figure 6.6).

The station types option in this menu allows the user to change the type of any existing station. The existing stations and their type are listed, and the user is asked to enter a station number (Figure 6.7). When he enters one of the existing station numbers, a menu of available station types is displayed, allowing the user to choose one. The last station is always the common storage, and the user is not allowed to change its type. To exit this option a zero should be entered for the station number.

Figure 6.6. Work Stations Update Menu

Figure 6.7. Changing the Station Types

Primary resource capacities may be changed by choosing the second option in the work stations update menu. The current capacities of the existing stations are displayed (Figure 6.8). The user is asked to enter a station number and the new primary resource capacity for the station. The primary resource capacity for a batch size independent process also defines the number of jobs that can be processed simultaneously at the station. Again a zero must be entered for the station number to exit this option.

In addition to the primary resource, two optional resources per station are allowed in generic models of IntelliSIM. The optional resources may be shared by multiple stations of the same type. The user may assign the first and the second optional resource types to work stations by choosing the third and the fourth options in the work stations update menu. Existing work stations and current optional resource assignments are displayed (Figures 6.9 and 6.10). The user enters the station number and the number of the optional resource assigned to the station. To distinguish between the first and the second set of optional resources, first optional resources are numbered starting with one, and second optional resources are numbered beginning with 51. To undo a optional resource assignment, a zero must be entered as the optional resource type in which case the station will not require a optional resource.

When the processing time distributions option is selected from the work stations update menu, another menu is displayed (Figure 6.11). This menu permits the user to define different distributions for the setup, processing, and machine loading/unloading operations at a station. When one of the three operations is selected, the station number is asked from the user, and the current distribution type is displayed (Figure 6.12). Then the user chooses one of the nine distributions provided in an additional menu.

Figure 6.8. Changing the Primary Resource Capacities

Figure 6.9. Changing the First Optional Resource Assignments

Figure 6.10. Changing the Second Optional Resource Assignments

Figure 6.11. Processing Time Distributions Menu

Figure 6.12. Changing the Processing Time Distributions

The efficiency ratios option under the processing time distributions menu allows the user to enter a different ratio for each station (Figure 6.13). Processing times will be adjusted accordingly during the simulation run.

Machining stations can be defined as alternative stations for other machining stations. When this option is selected, only the machining stations are listed including their alternatives (Figure 6.14). A zero for an alternative station means that there are no alternative stations for that station. A job is sent to an alternative station when the next regular station in the operations routing of the job is not available.

Failure rates for inspection stations are defined by choosing the last option in the work stations update menu. Only the inspection stations are listed with the current failure rates (Figure 6.15). The user enters the station number and the new failure rate for the station.

6.3. Work-In-Process Storage Areas

This option is concerned with the characteristics of work-in-process (WIP) storage areas and other queues in the system, such as queues for optional resources, pallets, fixtures, and material handling equipment. When this option is selected the WIP storage areas menu appears (Figure 6.16). The first option in this menu is to specify input buffer capacities (WIP storage capacities for incoming jobs that are waiting for processing). Current input buffer capacities are listed for the existing stations (Figure 6.17). The user is asked to enter a station number and the new buffer capacity. To exit this option a zero should be entered as the station number. The default buffer capacity is 9999 for all stations in a generic model.

Figure 6.13. Changing the Efficiency Ratios

Figure 6.14. Changing the Alternatives for Machining Stations

Figure 6.15. Changing the Failure Rates for Inspection Stations

Figure 6.16. WIP Storage Areas Menu

Figure 6.17. Changing the Input Buffer Capacities

The output buffer capacities (WIP storage capacities for outgoing jobs that finished processing and are waiting to be transferred to the next station) are defined by choosing the second option. As in input buffer capacities, current output capacities are listed, and new capacities are entered for selected stations (Figure 6.18). The default value for output buffer capacities is also 9999. Only the generic models whose names start with characters GQ1 have the output buffer capacities. If the user selects this option with one of the other generic models, a message will appear indicating that no output buffers exist (Figure 6.19).

Jobs arriving at a station when the input buffer capacity of the station is exceeded will be treated according to the balking option selected by the user (Figure 6.20). Generic models provide three options for this purpose: (1) Jobs will wait on transportation media until there is room in the station WIP area, (2) Jobs will block the previous station, and the primary resource of the previous station will not be released until the WIP area of the next station becomes available, (3) Jobs will be transferred to the common storage where they will wait until the WIP area of the next station becomes available. The default balking option is wait on transportation media.

The priority rule for removing jobs from the WIP storage areas is specified by choosing the priority rule option from the WIP storage areas menu. A menu of eight different priority rules will be displayed (Figure 6.21), and the user will be able to select one of them. The same priority rule will be used for all the WIP areas and other queues in the model. The default priority rule is FIFO (first-in-first-out).

Figure 6.18. Changing the Output Buffer Capacities

Figure 6.19. Attempting to Change the Output Buffer
Capacities When They Do Not Exist

Figure 6.20. Balking Options Menu

Figure 6.21. Priority Rules Menu

6.4. Optional and Special Resources

This option allows the user to define resource capacities for optional resources that are assigned to work stations and two special resource types, pallets and fixtures. The menu for this option is shown in Figure 6.22.

Only the generic models whose names include S00, S02, S12, or S04 as the last three characters have optional resources. These characters define the usage pattern of the two optional resource types which are described in the model status summary screen (Figure 6.1). If the currently selected model does not make use of optional resources, a message indicating this will appear when the first or the second optional resources option is selected (Figure 6.23). Otherwise, when the first optional resources option is selected, the current resource capacities are listed for existing optional resources (Figure 6.24). The user enters a optional resource number and the new capacity for the resource. To exit this option a zero should be entered for the resource number. The capacities of the second set of optional resources are updated in a similar manner (Figure 6.25). The default value for optional resource capacities is one.

There are as many first optional resource types and second optional resource types as the number of work stations. Whenever a new station is added by the user, IntelliSIM automatically adds a new first optional resource type as well as a new second optional resource type. This way the user may assign a different optional resource type to each of the work stations (see Section 6.2.2 for optional resource assignments to stations), in which case optional resources are dedicated to stations. If a optional resource is shared by a number of stations, then this optional resource type may be assigned to multiple stations. The only restriction on this assignment is that a optional resource may be shared by the stations of the same type, e.g., by machining stations or by inspection stations. The same resource may not be assigned to a machining and an inspection station at the same time.

Figure 6.22. Optional and Special Resources Menu

Figure 6.23. Attempting to Change the Optional Resource
Capacities When They Do Not Exist

Figure 6.24. Changing the First Optional Resource Capacities

Figure 6.25. Changing the Second Optional Resource Capacities

In addition to the optional resources, there are two special resources in generic models, pallets and fixtures. Pallets are used for transportation purposes only. They are allocated to a job after processing of the job is finished at a station, remain seized during transportation of the job to next station, and released as soon as the job arrives at the next station. Fixtures, on the other hand, are allocated to jobs at fixture loading/unloading stations. A job remains on fixtures as it visits a series of stations for different processes, and fixtures are released only when the job is routed back to a fixture loading/unloading station for unloading.

There are two types of pallets in generic models. The capacity (total number of available units) of each pallet type can be changed as in Figure 6.26. The default value for pallet capacities is zero which indicates that no pallets are used. Pallets are assumed to be used when their capacity is set to a nonzero value. When pallets need not be used, their capacities must be set back to zero. The number of parts that can be loaded on each pallet type is specified as in Figure 6.27. The default value for number of parts per pallet is one for both pallet types.

Two fixture types are defined in generic models. The capacities of two fixture types and the number of parts that can be loaded on them are updated in a similar manner (Figures 6.28 and 6.29). The default value for fixture capacities is 100, and the default for number of parts per fixture is one for both fixture types.

Figure 6.26. Changing the Pallet Capacities

Figure 6.27. Changing the Number of Parts Per Pallet Type

Figure 6.28. Changing the Fixture Capacities

Figure 6.29. Changing the Number of Parts Per Fixture Type

6.5. Material Handling System

There are three ways of transporting jobs between the stations as seen in the material handling system menu (Figure 6.30). These are manual transportation, transporters, and conveyors. The first option, manual transportation, implies that the jobs are assumed to be transported without the need of special material handling equipment such as agvs, carts, cranes, or conveyors. For this option, a distribution type and parameters are specified by the user to estimate the transportation time between station pairs (Figure 6.31). The default distribution is uniform.

Material handling equipment such as agvs, cranes, and carts can be modeled as transporters. Two different transporter types are defined in generic models as seen in the transporters menu (Figure 6.32). The capacity (number of transporter units available) and velocity of the first transporter type are updated as in Figure 6.32. The characteristics of the second transporter type are specified in the same way. The default values for transporter capacity and velocity are one for both transporters.

The third option in the transporters menu allows the user to define distances between the station pairs. When the user enters a starting station number (Figure 6.33), the distances from this station to other stations whose numbers are larger than the starting station are listed (Figure 6.34). The user enters the number of one of the listed stations and the new distance from the starting station to this station. For example, if the station number two is the starting station, then distances between station pairs 2-3, 2-4, 2-5, and so on can be updated. Distances between station pairs 3-2, 4-2, and 5-2 are the same as the distances between pairs 2-3, 2-4, and 2-5, respectively. Distances for all station pairs are one by default. The units of transporter velocity and distances must be compatible, e.g., if the distances are defined in feet, then the velocity must be defined in feet per simulation time unit.

Figure 6.30. Material Handling System Menu

Figure 6.31. Changing the Manual Transportation Distribution

Figure 6.32. Transporters Menu and
Changing the Features of Transporters

Figure 6.33. Entering the Starting Station for Transporter Distances

Figure 6.34. Changing the Transporter Distances

The third type of transportation media are conveyors. Two different conveyor types are defined in generic models (Figure 6.35). The cell length and velocity of the first conveyor are specified as in Figure 6.35. These characteristics are entered in the same way for the second conveyor type. The default values for cell length and velocity are one for both conveyors. As in transporters, the units of conveyor cell length and velocity must be compatible, e.g., if the cell length is defined in feet, then the velocity must be defined in feet per simulation time unit. The path for the first conveyor is specified by selecting segments option for the first conveyor type. A conveyor segment is part of the conveyor from one work station to another, and a number of segments constitute the conveyor.

The first "from" station for the conveyor is entered as in Figure 6.36. This station is the first station from which the conveyor starts. Then a "to" station and a distance are entered defining the first segment (Figure 6.37). This "to" station automatically becomes the "from" station of the second segment, the "to" station of the second segment becomes the "from" station of the third segment, and so on. The conveyor segment specification ends when the user enters a zero for the "to" station. Segments for the second conveyor are defined in a similar way and may be completely different from the first conveyor. The default segments for both conveyors defined by station pairs are 1-2, 2-3, 3-4, 4-5, 5-6, 6-1, and default distances are all one. The default segments are circular and any station is accessible from any other directly or indirectly.

Figure 6.35. Conveyors Menu and
Changing the Features of Conveyors

Figure 6.36. Entering the Starting Station for Conveyor Segments

Figure 6.37. Changing the Conveyor Segments

6.6. Jobs

A job is a batch of parts visiting a number of work stations for various processing steps. A job type is distinguished by its operation routing (sequence of stations that it visits for processing). There may be multiple batches of the same job type flowing through the system at any time. When the user chooses jobs option from the model configuration menu, the jobs menu is displayed (Figure 6.38). There are three options in this menu for addition, update, and deletion of jobs.

6.6.1. Addition of a New Job Type

The first option in the work stations menu allows addition of a new job type. Up to fifty job types may be added to a generic model. Each time the user selects this option a new job type is added with the following default values.

- Number of batches for arrival patterns A0 and A3: one.
- Arrival times of batches for arrival pattern A3: zero.
- Due dates of batches for arrival patterns A0 and A3: zero.
- Time between arrivals distribution for arrival pattern A1 and A2: uniform with minimum and maximum values of ten.
- Cumulative percentage of job type for arrival pattern A1: 100 percent.
- Pallet type: one.
- Fixture type: one.
- Batch size: one.
- User assigned priority: zero.
- Match numbers for assembly: zero.
- Operation routing: Only first station is visited.
- Transportation between stations: Manual.
- Processing time distribution parameters (setup, process, loading/unloading): zero.

Figure 6.38. Jobs Menu

6.6.2. Update of Existing Job Types

The menu to update the existing jobs is shown in Figure 6.39. The first option in this menu is concerned with the specification of job arrivals. When the job arrivals option is selected, IntelliSIM takes different actions depending on the arrival pattern of the currently selected generic model. Job arrival specification for each of the four arrival patterns is described below.

A0 - All jobs are available at the beginning of the simulation period

All batches of all job types are available at the beginning of the simulation period with this arrival pattern. The user needs to specify the number of batches for each job type. The job number and the number of batches for the job are entered as in Figure 6.40. The maximum number of batches allowed is 999 for each job type.

A1 - Job types are randomly specified as percentages of random arrivals

When job arrivals option is selected with this option, the menu in Figure 6.41 appears. The first option in this menu is to choose a distribution and specify its parameters to generate interarrival times of jobs (Figure 6.42). The second option allows the user to define cumulative percentages for job types. Job type of each arrival generated from the selected distribution will be defined depending on these percentages. In Figure 6.43, on the average 20 percent of the arrivals will be of job type one, 10 percent job type two, another 10 percent job type three, 20 percent job type four, 25 percent job type five, and 15 percent job type six. To exit this option a zero must be entered for the job number.

Figure 6.39. Jobs Update Menu

Figure 6.40. Specification of Number of Batches

Per Job Type for Arrival Pattern A0

Figure 6.41. Job Arrivals Menu for Arrival Pattern A1

Figure 6.42. Choice of Job Arrival Distribution for Arrival Pattern A1

Figure 6.43. Specification of Cumulative Percentages
of Job Types for Arrival Pattern A1

A2 - Random interarrival times are specified for each job type

Another job arrival menu is displayed with this arrival pattern (Figure 6.44). The first option in this menu is for selecting a common job arrival distribution. The user enters different arrival parameters for each job type by choosing the second option (Figure 6.45). The job number is entered, current parameter values are displayed, and the user is asked to enter the new parameters for the selected job. To exit this option a zero must be entered for the job number.

A3 - Deterministic arrival times are specified for each batch of each job

Deterministic arrival times may be specified for each batch of each job type with this arrival pattern. There are two options in the job arrivals menu (Figure 6.46). The first option is to specify the number of batches for each job type. The user enters a job number and the number of batches for the job (Figure 6.47). A zero should be entered for the job type to exit this option. Deterministic arrival times are specified by choosing the second option in the menu. When this option is selected, the user enters a job number, and constant arrival times of all batches of the selected job type are listed (Figure 6.48). The user may enter a batch number and the new arrival time for the batch. The arrival time of a batch must always be larger than that of preceding batch. To exit this option a zero must be entered for the batch number.

Figure 6.44. Job Arrivals Menu for Arrival Pattern A2

Figure 6.45. Specification of Job Arrival
Distribution and Parameters for Arrival Pattern A2

Figure 6.46. Job arrivals menu for Arrival Pattern A3

Figure 6.47. Specification of Number of Batches

Per Job Type for Arrival Pattern A3

Figure 6.48. Specification of Arrival Times of
Batches for Arrival Pattern A3

Pallet types for different job types may be changed by choosing the second option in the jobs update menu. The current pallet types for each job are displayed (Figure 6.49). The user is asked to enter a job number and the new pallet type for the job (either one or two). A zero must be entered for the job number to exit this option. Fixture types per job type may be changed in a similar manner (Figure 6.50).

As mentioned before, each job entity in a generic model represents a batch of parts of the same type. The user should choose the batch sizes option from jobs update menu to specify the batch sizes for each job type. The current batch sizes are displayed for the existing jobs (Figure 6.51). The user is asked to enter a job number and the new batch size for the job. A one may be entered as the batch size in which case each job entity represents a single part rather than a batch. A zero must be entered for the job number to exit this option.

User assigned priorities are updated by selecting the fifth option in the jobs update menu. These priorities are used when the user assigned priorities option is chosen as the priority rule for work-in-process storage areas. (Priority rule selection is explained in section 6.2.2.) When this option is selected, the current priorities are displayed for the existing jobs (Figure 6.52). The user is asked to enter a job number and the new priority for the job. A zero must be entered for the job number to exit this option.

Figure 6.49. Changing the Pallet Types

Figure 6.50. Changing the Fixture Types

Figure 6.51. Changing the Batch Sizes

Figure 6.52. Changing the User assigned Priorities

Due dates for batches may be specified when a generic model with arrival pattern A0 or A3 is selected. Arrival times of batches are zero or constant with these two arrival patterns, hence deterministic due dates may be defined for batches. When this option is selected, the user selects a job type, and the current due dates are displayed for the batches of this job (Figure 6.53). The user is asked to enter a batch number and the new due date for the batch. A zero must be entered for the batch number to exit this option. With arrival patterns A1 and A2, arrival times are random, and due dates cannot be defined. When due dates option is chosen with arrival patterns A1 and A2, a message is displayed indicating that due dates can not be defined (Figure 6.54).

Match numbers are assigned to batches to decide which batches to merge at assembly stations. Different match numbers can be defined for each batch of each job with arrival patterns A0 and A3, because the number of batches is known for each job type. For example the second batch of job type one may be matched with the third batch of job type two to form a subassembly. When this option is selected, the user selects a job type, and the current match numbers are displayed for the batches of this job (Figure 6.55). The user is asked to enter a batch number and the new match number for the batch. A zero must be entered for the batch number to exit this option. Batches that are matched must have the same batch sizes.

Batch arrivals are randomly generated with arrival patterns A1 and A2, and the number of batches per job type is not known. Therefore, with these arrival patterns, match numbers are defined for job types rather than batches. When match numbers option is selected, current match numbers are displayed for the existing jobs (Figure 6.56). The user is asked to enter a job number and the new match number for the job. A zero must be entered for the job number to exit this option. Again, jobs that are matched must have the same batch sizes.

Figure 6.53. Changing the Due Dates for Arrival Patterns A0 and A3

Figure 6.54. Attempting to Change the Due Dates
for Arrival Patterns A1 and A2

Figure 6.55. Changing the Match Numbers for Arrival Patterns A0 and A3

Figure 6.56. Changing the Match Numbers for Arrival Patterns A1 and A2

6.7. Operation Routings

This option is for specification of the processing steps for jobs. The user may define the sequence of work stations each job type is supposed to visit to finish processing. Also the processing time parameters for machine setup, actual part processing, and machine loading/unloading can be specified. Before describing each of the options in the operation routings menu, the following rules must be considered in defining operation sequences.

- Three distribution types are specified per work station for generating setup, loading/unloading, and processing times, i.e., distribution type is station dependent. (See Section 6.2.2 for specification of processing time distributions.) Distribution parameters may be different for each operation of a job, i.e., parameters are operation dependent.
- Setup times are defined per batch, loading/unloading and processing times per part. Hence the total time for a batch is computed by multiplying the time per part (twice the loading/unloading time plus the processing time) by the batch size, and adding the setup time to this amount.
- Setup times are sequence independent.
- Service times for batch size independent processes must be specified with the setup time parameters.
- Service time may be zero for a particular suboperation. If the machine setup or machine loading/unloading is to be ignored, service time parameters for that suboperation must be set to zero.
- Efficiency ratios may be defined for each station. The standard service times generated from the distributions are adjusted accordingly. (If the efficiency ratio is less than one for a particular station, actual service times will be longer than the standard times.)
- A maximum of fifty operations per job type are allowed.

-
- One of the two jobs matching at an assembly station is regarded as the minor job, and the assembly station must be the last station in its operation sequence. The other job is major, and it may have additional stations in its routing following the assembly. The subassembly takes the job type of the major job.
 - Whenever an inspection station is specified in an operation sequence, the immediate successor must be a rework station (one of the machining stations). Only the jobs failing inspection are sent to the rework station. Jobs passing inspection skip the rework, and are routed to next station in the sequence.
 - A job visiting a fixture loading/unloading station to get loaded onto fixtures must be routed back to the same or a different loading/unloading station to get unloaded prior to leaving the system. Otherwise the fixtures that are allocated to the job will not be released.

The above rules must be followed when updating operation routings by choosing one of the options in the menu (Figure 6.57). The first option in this menu is definition of a new routing for an existing job. When this option is selected, the user selects a job type. In the following screen the user enters a sequence of work stations which the job is supposed to visit (Figure 6.58). A zero must be entered for the station number to end the sequence. Given this sequence, IntelliSIM starts asking for the processing times. For each station the job visits, the current machine setup, process, and machine loading/unloading distributions are displayed, and the user is asked to enter the distribution parameters for these three suboperations. As an example, Figure 6.59 shows the entry of the setup time parameters for the fourth operation of job type one. The operation is performed at station two, and the setup distribution is uniform.

Figure 6.57. Operations Routing Menu

Figure 6.58. Defining a New Operation Routing for a Job

Figure 6.59. Specifying Processing Time Parameters
for a New Operation Routing

The second option in the operation routings menu is for adding an operation to an already defined operation routing. When the user selects a job type, the current operation sequence for the job is displayed (Figure 6.60). Then the user enters an operation number, the station number at which this operation will be performed, and the processing time parameters for three suboperations. A zero must be entered for the operation number to exit this option. The number of the additional operation may be the last existing operation plus one, in which case the new operation will be appended to the end of the sequence. If an already existing operation number is entered for the operation to be added, the new operation will be inserted right before the existing one. Operation numbers of the succeeding operations will be increased by one.

The third option in the operation routings menu is for deleting an operation from an already defined operation routing. When the user selects a job type, the current operation sequence for the job is displayed (Figure 6.61). Then the user enters an existing operation number which will be deleted. Operation numbers of the succeeding operations will be decreased by one. A zero must be entered for the operation number to exit this option.

The change operations option in the operation routings menu is concerned with changing various features of existing operations. The user may choose to change the station numbers, setup time parameters, process time parameters, or loading/unloading time parameters from the menu (Figure 6.62). When the user selects one of these options, he enters a job type. The current operation routing is listed for the job, an operation number is entered, and the station number or processing time parameters are changed according to the user's change option. The update of the process time parameters for the second operation of job type one is shown in Figure 6.63 as an example. A zero must be entered to exit the operation routing change options.

Figure 6.60. Adding a New Operation to an Existing Routing

Figure 6.61. Deleting an Operation from an Existing Routing

Figure 6.62. Change Operations Menu

Figure 6.63. Changing Process Time Parameters
for an Existing Operation

The last option in the operation routings menu is for specification of material handling equipment type between operations. When the user chooses this option he enters a job number. The current operation routing of the job and the material handling equipment types are listed (Figure 6.64). The user enters an operation number, and selects the new equipment type from the transportation equipment menu. The equipment type for a particular operation indicates that the job will be transferred with this equipment type from the current station to the following station. If the material handling equipment type is blank, then the job will continue to use the latest specified equipment type. This way the user does not have to specify the equipment type for each and every operation unless it changes.

With manual transportation and transporters, any station can be accessed from any other. This is not necessarily true for conveyors. When one of the conveyors is selected for a particular operation, the user has to make sure that the following station is accessible from the current station by the way conveyor path is defined (See Section 6.5 for specification of conveyor segments). For example, in Figure 6.64 job number one is supposed to be transferred from station five to station two by conveyor type one. The segments defined for conveyor type one should allow this transfer, directly or indirectly. A segment may be defined directly from station five to station two, or a number of successive segments defined for station pairs 5-3, 3-6, 6-2 allow indirect access from station five to station two.

6.8. Equipment Breakdown/Maintenance

The equipment breakdown/maintenance option in the model configuration menu allows the user to define breakdowns and scheduled maintenance activities for primary station resources and material handling equipment. The equipment breakdown/maintenance menu is shown in Figure 6.65. The first four options in this menu are for selecting distribution types for different purposes.

Figure 6.64. Changing Material Handling Equipment Type

Figure 6.65. Equipment Breakdown/Maintenance Menu

-
- Time between failures distribution is for generating interarrival times between equipment breakdowns.
 - Time to finish repair distribution generates the time elapsed to repair an equipment when a breakdown occurs.
 - Time between maintenance distribution is used to generate the interarrival times between scheduled maintenance activities.
 - Time to finish maintenance distribution generates the time that the maintenance takes.

Choice of the the time between failures distribution is given in Figure 6.66 as an example. One of the nine distributions provided in the menu may be selected for each of the above options in a similar manner. Default distributions are uniform.

The fifth option in the equipment breakdown/maintenance menu is to specify which station primary resources and material handling equipment types should have breakdowns. When this option is selected, existing work stations and four material handling equipment types are listed (Figure 6.67). An asterisk (*) for a station or material handling equipment indicates that currently breakdowns are defined for that station or equipment. The user enters a station number, and decides if he wants to define breakdowns for the station. If breakdowns are to be defined, he enters the time between failure distribution parameters for the selected station (Figure 6.68). Time to finish repair distribution parameters are entered in a similar manner. To exit this option, a zero should be entered as the station number. By default, no breakdowns are defined for stations or material handling equipment.

Scheduled maintenance activities are defined by choosing the last option in the equipment breakdown/maintenance menu in a similar manner (Figure 6.69). Time between maintenance and time to finish maintenance distribution parameters are defined for those stations which have scheduled maintenance (Figure 6.70). By default, no maintenance activities are defined for work stations or material handling equipment.

Figure 6.66. Changing Time Between Failures Distribution

Figure 6.67. Specification of Equipment Breakdowns

Figure 6.68. Specification of Time Between Failures Distribution Parameters

Figure 6.69. Specification of Scheduled Maintenance

Figure 6.70. Specification of Time Between Maintenance Distribution Parameters

6.9. Fit Data Distribution

This option allows the user to enter a set of raw data, such as machine setup times for a particular operation of a job, and test the data to see if the data fits a certain probabilistic distribution. This feature of IntelliSIM can be used to decide the types and parameters of distributions that are needed for model configuration. When the raw data are available, but the user does not know what distribution they come from, he can enter the data using a data fitting editor and test the data for distribution fitting.

The fit data distribution menu includes three options (Figure 6.71). The first option in this menu provides the user with an editor to write data to a file (Figure 6.72). Each datum must be on a separate line. When data entry is complete, the user presses F10 key to save the data on his floppy disk. The data are saved under a file name specified by the user.

The second option in the fit data distribution menu is used to test the data saved in a file for distribution fitting. When the user selects this option, IntelliSIM displays information on-screen and asks the user to enter the name of the file (Figure 6.73). At this point, the user is provided an option of viewing the data file. If the user answers **yes**, the numbers in the data file will be displayed. At this time, IntelliSIM provides the user with general statistics for the data (Figure 6.74). IntelliSIM also provides the user with the option of viewing the feasibility values used within the distribution fitting procedure. IntelliSIM performs a goodness of fit test. The user is provided with each feasible distribution and a conclusion on its acceptance or rejection (Figure 6.75). IntelliSIM selects the best distribution and supplies the parameter values for that distribution (Figure 6.76).

The third option under fitting a data distribution is **New Distribution Fitting Output**. With this option, IntelliSIM displays all the details of the distribution fitting that was done in an editor (Figure 6.77). The user can save the output in a file.

Figure 6.71. Fit Data Distribution Menu

Figure 6.72. Data Fitting Editor

Figure 6.73. Data File Name Prompt

Figure 6.74. General Statistics of Data Set

Figure 6.75. Feasibility Rules

Figure 6.76. Data Fitting Conclusion

Figure 6.77. Data Fitting Summary Report

As an example, Kolmogorov-Smirnov goodness of fit test is applied to a set of data to test for normality. The observed and theoretical cumulative frequencies and the Kolmogorov-Smirnov test statistic are computed (Figure 6.76). Then, the test hypotheses are stated, and test results are displayed for different confidence levels of alpha (Figure 6.77).

6.10. Output Statistics

This option allows the user to specify some of the output statistics that will be generated by simulation. There are two types of statistics in generic models. The first type of statistics are the observation-based (tally) statistics that are generated automatically at the end of each run. These statistics provide information regarding the time spent in the system for all job types. The second type of statistics are the time-persistent (dstat) statistics which summarize resource utilizations and jobs waiting in the work-in-process storage areas and other queues. Not all time-persistent statistics are generated automatically because of the large memory requirements. Instead, the user selects the statistics that he wants to be generated.

Selection of time-persistent statistics to be generated is done through a menu shown in Figure 6.78. Statistics for work stations and statistics for optional resources are specified separately by choosing one of the two options in this menu. Time persistent statistics for work stations are specified as in Figure 6.79. The user enters a station number and decides if he wants the statistics for the station to be generated. To exit this option, a zero must be entered for the station number. By default, all time persistent statistics are generated for existing stations. The statistics for a station include the following.

- Station utilization.
- Number of jobs waiting in the input buffer.
- Number of jobs waiting in the output buffer (if any).
- Number of jobs waiting to be merged in assembly stations (if any).

Figure 6.78. Output Statistics Menu

Figure 6.79. Specifying Statistics for Work Stations

Time persistent statistics for optional resources are specified as in Figure 6.80.

The user enters a optional resource number and decides if he wants the statistics for the resource to be generated. To exit this option, a zero must be entered for the resource number. By default, no time persistent statistics are generated for optional resources. The statistics for a optional resource include the optional resource utilization and the number of jobs waiting for the resource.

In addition to the user-selected statistics for stations and optional resources mentioned above, the following time-persistent statistics are generated automatically.

- Utilization and Number of jobs waiting in the queue for transporter type one.
- Utilization and Number of jobs waiting in the queue for transporter type two.
- Utilization and Number of jobs waiting in the queue for conveyor type one.
- Utilization and Number of jobs waiting in the queue for conveyor type two.
- Utilization and Number of jobs waiting in the queue for pallet type one.
- Utilization and Number of jobs waiting in the queue for pallet type two.
- Utilization and Number of jobs waiting in the queue for fixture type one.
- Utilization and Number of jobs waiting in the queue for fixture type two.
- Number of jobs waiting in the common storage.
- Number of jobs waiting on the material handling equipment.
- Number of jobs blocking the stations.

Figure 6.80. Specifying Statistics for Optional Resources

6.11. Model Status Summary

When the user chooses the model status summary option from the model configuration menu, the model status summary appears on the screen (Figure 6.1). If the currently selected model is generic, this screen provides a summary of the default configuration of the generic model. For a specific model, the latest configuration is displayed as it was saved by the user. The summary screen provides a quick review of some of the specifics of the model. When the user presses any key the summary screen disappears, and the user is taken back to the model configuration menu.

6.12. Exit Model Configuration

When the user wants to exit model configuration, he may do so by selecting the last option in the model configuration menu, or by pressing the <Esc> key when the model configuration menu is on the screen. If the user quits by pressing the <Esc> key, the model will not be saved. Otherwise, IntelliSIM requires the user to provide the name of the specific model.

CHAPTER 7

EXECUTE SIMULATION

This option is concerned with the execution of the specific models. In order to use the execute simulation modules of IntelliSIM, the user must have at least one specific model generated in SIMAN format (a .EXP file), and saved on a floppy disk. Execute simulation modules require a specific .EXP file structure, hence not every .EXP file can be used for executing simulation with IntelliSIM. The .EXP files in this specific structure are generated automatically by the model configuration module of IntelliSIM when the user chooses the exit model configuration option which is discussed in Section 5.12.

Alternatively, the user may develop a .EXP file manually, that is associated with one of the generic models in IntelliSIM. This, however, can be done by an expert user only, and the nonexpert user is encouraged to use IntelliSIM for model configuration and automatic generation of .EXP files.

When the execute simulation option is selected from the main menu, the execute simulation menu is displayed (Figure 7.1). There are four options in this menu which are discussed in the following sections.

Figure 7.1. Execute simulation Menu

7.1. Manual Execution of Specific Model

This option allows the user to execute his specific model without any control by IntelliSIM on the number of simulation runs and the run length. The user specifies the number of runs and the length of each run when manual execution option is selected.

When the user chooses this option, IntelliSIM prompts him to insert a floppy disk in A drive which contains one or more specific models (Figure 7.2). The specific models (.EXP files) available on the floppy disk are listed in a menu so that the user may select one of them (Figure 7.3). The user may press the <Esc> key if he does not wish to execute any of the models in this menu, and wants to go back to the execute simulation menu.

After specific model is selected, IntelliSIM requires the user to enter number of runs and the run length(Figure 7.4). The selected specific model is automatically compiled, linked, and executed as many times and as long as specified by the user. The standard SIMAN summary report of output statistics is generated and displayed to the user at the end of the run. The summary report is also saved under the name SUMMARY.DOC in the directory C:\ISIM\USER. A sample summary report generated by manual execution is shown in Figure 7.5. The user can view, edit and save this summary report by selecting view and edit summary report option from execute simulation menu.

Figure 7.2. IntelliSIM Prompt for Specific Models Disk
to Select a Model for Manual Execution

Figure 7.3. Specific Models Menu to Select a Model for Manual Execution

Figure 7.4 Simulation run length and number of runs for a specific model

Figure 7.5. Sample SIMAN Summary Report Generated
by Manual Execution of a Specific Model

7.2. Semi-Automatic Analysis

This option allows the user to execute the specific model selected with the confidence interval and percent error for the selected priority variable displayed on the screen in real time. When the user selects this option, IntelliSIM requires the user to specify type and name of the priority variable. The type of priority variable can be DSTAT or TALLY - time based or observation based. Once user selects the type of priority variable (Figure 7.6) , IntelliSIM displays list of all the available variables of that type for a particular model. User now selects priority variable from that list (Figure 7.7) .

In IntelliSIM, batch size under this option is not under the control of user. After the selection of priority variable, actual execution of the particular model starts. The confidence interval and percentage error for the priority variable is displayed on the screen at the completion of every one batch (Figure 7.8) . The user can quit the semiautomatic analysis at any time by pressing the END key. Also, if percent error reaches less than 5%, then IntelliSIM automatically quits the semiautomatic analysis. At the end of semiautomatic analysis, IntelliSIM displays the results of confidence interval building and percent error graphically so as to aid the user in making decisions (Figure 7.9) .

The user can view,edit and save the summary report generated by selecting view and edit summary report option. This summary report is a standard SIMAN summary report and the table showing confidence interval and percent error for the priority variable (Figure 7.10) .

Figure 7.6. - Selection of type of priority variable

Figure 7.7 - Selection of the name of priority variable from the list.

Figure 7.8 - Table showing the confidence interval and percent error for the priority variable in real time in semiautomatic analysis

Figure 7.9. - Graphical display of results at the end of semiautomatic analysis

Figure 7.10. - Sample summary report generated in the semiautomatic analysis

7.3. Viewing and Editing the Summary Report

This option allows the user to view, edit, and print the Summary Report generated by a completed simulation. The editor works like most other editors; the user can use the page-down, page-up, and arrow keys for moving around the Summary Report. The option View Summary Report does not allow the user to make modifications; to do so use the Edit Summary Report option.

CHAPTER 8

ADDITION OF NEW MODELS IN INTELLISIM MODEL BASE

This section is not for all the users of IntelliSIM. An expert modeler is needed to be able to add new models in the IntelliSIM model base. He must be an expert programmer in SIMAN and Turbo Prolog, as well as being familiar with the IntelliSIM concepts and programming structure.

8.1. File Format for Specific Models

A SIMAN model consists of a model structure (.MOD file) and an experimental frame (.EXP file). Generic models in IntelliSIM are .MOD files stored under the directory C:\INTELLISIM\MODELS. A .EXP file is generated by IntelliSIM when configuration of a generic model is complete. This file carries the name of the generic model structure it is associated with.

As the model configuration proceeds the changes are stored in Prolog format rather than SIMAN format, because the model configuration program is written in Prolog. A generic experimental frame which contains all the elements necessary to configure a generic model is stored as a Prolog database file namely GENERIC.FAC. This file carries the default values for all data items. As soon as the model configuration option is chosen, a copy of this file (GENERIC.FAC) is created under the name TEMP.FAC, and is used as a template for configuration. The name of the currently selected generic model is also recorded in this file. As the configuration proceeds, the data entered and changes made are stored in TEMP.FAC. When the user wants to leave the model configuration, he may choose to discard TEMP.FAC, or save it on a floppy disk under a name he specifies with the file extension .FAC. This user-named file, including the name of the generic model it is associated with, constitutes the specific model. The user may leave model configuration any time after saving his specific model, i.e., he does not have to finish

configuration at once. He may later select the same specific model and continue with the configuration.

When the user completes model configuration and wants to execute his specific model, the configuration module automatically reads the specific model (.FAC file), and generates an experimental frame (.EXP file) in SIMAN format. This file is compiled and linked with the generic model structure (.MOD file) for which it is created. The specific model in SIMAN format can be run by the post processing modules of IntelliSIM.

The Prolog database file GENERIC.FAC, which is used as a template for the experimental frame, and an example of the experimental frame (.EXP file) generated at the end of the model configuration are provided in the Technical Manual.

8.2. Addition of New Generic Models for Preprocessing

Addition of a generic model in IntelliSIM means addition of a new model structure (.MOD file). The model structure must meet certain conditions if the user wants to be able to use IntelliSIM to configure this new model.

Use of experimental frame elements in the new .MOD file must be in accordance with the format of GENERIC.FAC file and resulting experimental frame generated by IntelliSIM. For example, parameter P(17,1) must be used in the new generic model for the same purpose as it is in the existing generic models. Only then IntelliSIM can be used to configure the new model and generate specific models.

To add a generic model in IntelliSIM model base, the user has to copy the new .MOD file under the subdirectory C:\ISIM\MODELS.

The name of the new .MOD file needs to be added to the generic model names menu in IntelliSIM.PRO, and IntelliSIM.PRO must be re-compiled after adding the model name.

The user may select the new model from the generic model names menu, and configure the model using IntelliSIM to create a specific model.

8.3. Addition of New Specific Models for Post Processing

Addition of a specific model in IntelliSIM means creating a model structure (.MOD file) on the hard disk, and an experimental frame (.EXP file) on a floppy disk.

The new .MOD file must be copied under the subdirectory C:\ISIM\MODELS. There is only one restriction on the structure of the new .MOD file. To use post processing modules of IntelliSIM, the simulation entities which represent jobs in the model structure must be sent to the EVENT subroutine just before they are disposed. This is done by the following EVENT block.

EVENT:999:DISPOSE;

The above EVENT statement must be inserted wherever the entities in the model are disposed. The user should get rid of other DISPOSE modifiers that might exist in the model prior to the EVENT block.

The .EXP file for the new model must be stored on a floppy disk. When a post processing option is selected, IntelliSIM provides a menu of the .EXP files available on the current floppy disk so that the user may select a specific model. These .EXP files may be created by the model configuration module of IntelliSIM, or the user may create his own experimental frames for newly added generic models. The .EXP files must meet the following specifications.

- All the experimental frame element names, such as PROJECT, REPLICATE, TALLIES, and DSTAT, must begin at the tenth column (see the Technical Manual for an example of the experimental frame generated by IntelliSIM for exact format requirements).
- PROJECT element of the experimental frame must carry the corresponding .MOD file name (maximum eight characters) as in the following example.

PROJECT, <model structure file name>.MOD,,;

- Only one TALLY or DSTAT statistic per line is allowed ending with a colon (or semi-colon for the last element).

- No file units must be associated with the TALLY and DSTAT statistics.

When the user selects a newly added .EXP for post processing, IntelliSIM will recognize the corresponding .MOD file from the PROJECT element and look for it under the subdirectory C:\ISIM\MODELS.