School of Operations Research
and Industrial Engineering
College of Engineering
Cornell University
Ithaca, New York

Technical Report No. 752
August 1987

COSMOS
The Cornell Simulator of Manufacturing Operations
Version 3.0
by
P.L. Jackson
J.A. Muckstadt
C.V. Jones

This research was supported in part by NSF Grant DMC–85–09131, and by grants from
General Foods Corporation, Emerson Electric, the New York State Science and Technology
Foundation, and the Cornell Manufacturing Engineering and Productivity Program.
COSMOS
User's Manual
Version 3.0
System Architecture
P.L. Jackson
J.A. Muckstadt
C.V. Jones

System Implementation
R. Ferstenberg
S. Bellantoni
C. Olling
W. Martin
G. Tsai

Documentation
H. Spector

Copyright (c) 1986-Cornell Research Foundation, Inc., All Rights Reserved.
# TABLE OF CONTENTS

Chapter 1  COSMOS

1.1 Introduction ................................................. 1-1

1.2 Revitalizing the Manufacturing Process ..................... 1-2
1.2.1 Design for Manufacturability .............................. 1-3
1.2.2 Quality .................................................. 1-3
1.2.3 Just-In-Time .............................................. 1-4
1.2.4 Cost Accounting and Engineering Economics .............. 1-5
1.2.5 Computer Integrated Manufacturing ....................... 1-5
1.2.6 Scheduling Complexity .................................. 1-6
1.2.7 Material Logistics and Design ........................... 1-7
1.2.8 Computer Aided Logistics ............................... 1-8
1.2.9 Conclusion ............................................... 1-8

1.3 Evolution of Computer Aided Logistics Modeling Systems ... 1-8
1.3.1 Programming Languages .................................. 1-8
1.3.2 Simulation Languages .................................... 1-9
1.3.3 Optimization Languages .................................. 1-9
1.3.4 Planning and Scheduling Languages ....................... 1-10
1.3.5 Expert Systems .......................................... 1-10
1.3.6 User Interfaces ......................................... 1-11
1.3.7 Framework for a CAL System ............................ 1-13
1.3.8 Summary .................................................. 1-14

1.4 The COSMOS Modeling System .................................. 1-14
1.4.1 Representation ............................................ 1-14
1.4.2 An Example ................................................ 1-15
1.4.3 The Product Structure Diagram ......................... 1-17
1.4.4 The Operation Task Chart ............................... 1-18
1.4.5 Optimization: Planning, Scheduling, and Dispatching ... 1-19
1.4.6 Cyclic Scheduling ...................................... 1-20
1.4.7 Safety Stock ............................................ 1-21
1.4.8 Evaluation ............................................... 1-22
1.4.9 The Inventory Plot ...................................... 1-23
1.4.10 The Resource Utilization Chart ......................... 1-24
1.4.11 The Backlog Plot ....................................... 1-24
1.4.12 The Shortfall Plot ..................................... 1-25
1.4.13 The Material Cycle Time Plot ......................... 1-25
1.4.14 Integration .............................................. 1-27

1.5 Conclusion .................................................. 1-28
Chapter 2  Basic Concepts in COSMOS

2.1  Tasks.................................................. 2-1
2.2  Task charts vs. Precedence Diagrams.................. 2-1
2.3  Task charts........................................... 2-2
2.4  Precedence Edges..................................... 2-3
2.5  Infinite Loops and Root Nodes.......................... 2-3
2.6  Material Flow......................................... 2-3
2.7  Transfer Batches...................................... 2-4
2.8  Batching................................................. 2-5
2.9  Information Flow...................................... 2-6
2.10  Inspection............................................. 2-7
2.11  Part Numbers........................................... 2-7
2.12  Unitized vs. Batch Numbers............................ 2-8
2.13  Resources............................................... 2-8
2.14  Resource Types and Zones............................. 2-10
2.15  Processing Continuity................................ 2-10
2.16  Changeover Times and Setup Tasks.................... 2-10
2.17  The Resource Chart................................... 2-11
2.18  Dispatching............................................. 2-11
2.19  Scheduling and Task Chart Simulation................ 2-12
2.20  Product Structure Diagram............................ 2-13
2.21  Static vs. Dynamic Scheduling........................ 2-14
2.22  Different Algorithms Require Different Data........ 2-15
2.23  Tasks, Operations, and Processes..................... 2-16
2.24  Importance of the User Interface..................... 2-17

Chapter 3  COSMOS User Manual Description

3.1  Overview.............................................. 3-1
3.2  Related Documents.................................... 3-1
   3.2.1  COSMOS Data Structure........................... 3-1
   3.2.2  COSMOS Flatfile Description..................... 3-1
   3.2.3  COSMOS Screen Manager............................ 3-1
3.3  Notes on this Manual.................................. 3-1

Chapter 4  General Graphical Editing Functions

4.1  Hardware Requirements................................ 4-1
4.2  Software Requirements................................ 4-1
4.3  COSMOS Overview..................................... 4-2
4.4  Using the 5080 Graphics "Mouse"....................... 4-2
4.5  Translating A View................................... 4-2
4.6  Scaling A View........................................ 4-2
4.7  Reordering Views..................................... 4-2
4.8  Deleting A View....................................... 4-2
4.9  The Active View...................................... 4-3
Chapter 5  Main Menu
5.1  Inventory ................................................. 5-1
5.2  Mat Cy Lo ................................................. 5-2
5.3  Mat Cy Up ................................................. 5-3
5.4  Net Inventory ............................................ 5-4
5.5  Back Log .................................................. 5-5
5.6  Shortfall .................................................. 5-6
5.7  Resc Util .................................................. 5-7
5.8  Lot Rel ..................................................... 5-7
5.9  Editing a Test Case ........................................ 5-8
5.10 Simulator ................................................ 5-8
5.11 Help ...................................................... 5-8
5.12 Exit ....................................................... 5-8

Chapter 6  Product Structure Diagram Menu
6.1  Pan Left ................................................... 6-1
6.2  Pan Right .................................................. 6-1
6.3  Pan Up ...................................................... 6-1
6.4  Pan Down ................................................... 6-1
6.5  Zoom In .................................................... 6-1
6.6  Zoom Out ................................................... 6-2
6.7  Reset ....................................................... 6-2
6.8  Res Chrt ................................................... 6-2
6.9  Task Chrt .................................................... 6-2
6.10 View Node .................................................. 6-2
6.11 Edit a Node ................................................ 6-2
6.12 Move a Node ............................................... 6-3
6.13 Del Node ................................................... 6-3
6.14 Goto Node .................................................. 6-3
6.15 Add Edge .................................................... 6-4
6.16 View Edge .................................................. 6-4
6.17 Edit Edge ................................................... 6-5
6.18 Del Edge .................................................... 6-5
6.19 Shipping .................................................... 6-6
6.20 Receiving .................................................. 6-6
6.21 Transit ..................................................... 6-6
6.22 Stock ...................................................... 6-7
6.23 Operation .................................................. 6-7
6.24 Label ....................................................... 6-8
6.25 Simulator .................................................. 6-8
6.26 Get Case .................................................... 6-8
6.27 Save ....................................................... 6-9
6.28 Rename ..................................................... 6-9
6.29 File ......................................................... 6-9
6.30 Help ....................................................... 6-9
6.31 Exit ....................................................... 6-9
Chapter 7  Task Chart Menu

7.1  Pan Left................................................. 7-1
7.2  Pan Right............................................. 7-1
7.3  Pan Up................................................. 7-1
7.4  Pan Down.............................................. 7-1
7.5  Zoom In................................................ 7-1
7.6  Zoom Out.............................................. 7-2
7.7  Reset.................................................... 7-2
7.8  View Node............................................. 7-2
7.9  Edit Node............................................. 7-2
7.10 Move Node........................................... 7-3
7.11 Del Node.............................................. 7-3
7.12 Goto Node........................................... 7-3
7.13 View Node........................................... 7-4
7.14 Edit Edge............................................ 7-4
7.15 Del Edge.............................................. 7-4
7.16 Draw.................................................... 7-5
7.17 Store................................................... 7-5
7.18 Setup................................................... 7-6
7.19 Process............................................... 7-6
7.20 Inspect............................................... 7-7
7.21 Move.................................................... 7-7
7.22 Push.................................................... 7-8
7.23 Pull..................................................... 7-8
7.24 Adding a Probabilistic Edge................. 7-9
7.25 Get Case............................................. 7-10
7.26 Save.................................................... 7-10
7.27 Rename............................................... 7-10
7.28 File.................................................... 7-10
7.29 Help.................................................... 7-10
7.30 Exit.................................................... 7-10

Chapter 8  The Resource Chart Menu

8.1  Pan Left............................................... 8-1
8.2  Pan Right............................................. 8-1
8.3  Pan Up................................................ 8-1
8.4  Pan Down............................................. 8-1
8.5  Zoom In.............................................. 8-1
8.6  Zoom Out............................................. 8-1
8.7  Reset.................................................... 8-1
8.8  View Node........................................... 8-2
8.9  Edit Node............................................. 8-2
8.10 Move Node........................................... 8-2
8.11 Del Node.............................................. 8-3
8.12 Add Node............................................. 8-3
8.13 Get Case............................................. 8-3
8.14 Save.................................................... 8-3
8.15 Rename............................................... 8-3
8.16 File.................................................... 8-4
8.17 Help.................................................... 8-4
8.18 Exit.................................................... 8-4
Chapter 9 COSMOS Data Editors

9.1 Editing Overview...........................................9-1
9.2 Operating Node Editor.................................9-1
  9.2.1 Screen Function Description......................9-1
  9.2.2 Field Descriptions................................9-1
9.3 Product Structure Edge Editor......................9-3
  9.3.1 Record Type Description..........................9-3
  9.3.2 Field Descriptions...............................9-3
9.4 Material Activity Record Editor....................9-4
  9.4.1 Record Type Description..........................9-4
  9.4.2 Field Descriptions...............................9-4
9.5 Inventory Record Editor..............................9-5
  9.5.1 Record Type Description..........................9-5
  9.5.2 Field Descriptions...............................9-6
9.6 Product Structure Diagram Label Editor............9-7
  9.6.1 Record Type Description..........................9-7
  9.6.2 Field Descriptions...............................9-7
9.7 Task Node Editor.......................................9-8
  9.7.1 Record Type Description..........................9-8
  9.7.2 Field Descriptions...............................9-8
9.8 State Transition Record Editor.....................9-12
  9.8.1 Record Type Description..........................9-12
  9.8.2 Field Descriptions...............................9-12
9.9 Task Link Editor.......................................9-13
  9.9.1 Record Type Description..........................9-13
  9.9.2 Field Descriptions...............................9-13
9.10 Resource Editor.......................................9-13
  9.10.1 Record Type Description..........................9-13
  9.10.2 Field Descriptions...............................9-14
9.11 Contents of Node Editor.............................9-15
  9.11.1 Record Type Description..........................9-15
  9.11.2 Field Descriptions...............................9-15
9.12 Resource Type Selection Screen.....................9-16
  9.12.1 Record Type Description..........................9-16
9.13 Resource Type Editing Screen.......................9-18
  9.13.1 Field Description.................................9-18

Chapter 10 The COSMOS Simulator

10.1 Background...............................................10-1
10.2 Field Descriptions...................................10-1
1.1 Introduction

Material logistics is the planning, scheduling and control of material flow through a manufacturing and distribution system. An efficient material logistics system can make an enormous difference in the profitability and competitive posture of a manufacturing company. However, the complexity of the material logistics problem, and ultimately the performance of the manufacturing-distribution system, is usually determined by the basic design of the system --product design, process design, and facility design. Consequently, a thorough understanding of material logistics issues is critical when designing or re-designing manufacturing systems. Computer aided logistics systems allow manufacturing engineers to observe closely the interactions between design and performance (in design studies), and to plan and control material flow in operational studies. This paper describes a prototypical computer aided logistics system called COSMOS, an acronym for the COrnell Simulator of Manufacturing OperationS. COSMOS differs from other such systems in that it integrates planning, scheduling and simulation studies around a common database structure and employs computer graphics to simplify the engineer’s interaction with the system.

Before describing the COSMOS system, a number of topics deserve mention. Thus, Section 2 is a brief survey of concepts that have been proposed for the revival of manufacturing competitiveness. In addition, Section 2 reviews trends that have shaped the development of computer aided logistics systems. Section 3 is a review of significant developments in manufacturing modeling and simulation. The last section of this chapter is an overview of the COSMOS system. Chapter 2 introduces the user to some basic concepts and terms in COSMOS. Finally, the remainder of the manual details the COSMOS menus and editors.
1.2 **Revitalizing the Manufacturing Process**

During the past decade, the environment faced by manufacturing companies has changed dramatically and permanently. Significant increases in international competition have had a devastating effect on many U.S.-based manufacturing companies. Traditional markets have been flooded with products developed and produced in foreign countries. The captive consumer base that U.S. companies enjoyed for two post-war decades has largely disappeared.

In some instances, domestic companies have attempted to exploit wage differentials between the U.S. and other countries by shifting production overseas. However, they have often been caught short by sudden shifts in currency exchange rates, thereby reducing or reversing initial cost advantages.

Competition has been mounted on all aspects of product cost (design, manufacture, delivery and service), on all measures of product value (functionality, performance, operating economy, reliability, and serviceability) and on all measures of service value (product availability, customer support, and repair downtime). Thus, international competitiveness will not be restored by merely shifting production to low wage rate countries, by securing wage concessions or by imposing tariffs to adjust for wage differentials.

The new competitive environment is characterized by many sundry characteristics. First, product variety has increased as markets have been continually redefined and refined. Secondly, product life has shrunk as the pace of technological innovation has quickened. Next, competitive lead times for design, manufacture, delivery, and repair have been shortening. And finally, capital investment requirements and profitability have grown as product and process complexity have increased.

In many cases, the response to these challenges has been one of contraction -- work force reductions, wage concessions, supplier concessions, tax concessions, asset disposal, plant closings and business sell-outs. Indeed, the process seems to have accelerated with the recent wave of highly leveraged take-overs; new corporate owners have moved to achieve major and immediate cost reductions. For those companies that have made no adjustment, the next economic recession will be a traumatic experience. However, some companies have gone beyond cost reduction, and have achieved structural changes in all aspects of their business operations. These companies have revised their perception of
the market and the customer, and have changed their management philosophy, organizational form and employee attitudes around this new perspective.

1.2.1 Design for Manufacturability

This metamorphosis has influenced the design of products, processes and manufacturing systems. Traditionally, product designers have given their designs to manufacturing with little understanding of the manufacturing problems induced by a design. Not only has this resulted in substantial cost overruns, but it has also created subsequent service problems and numerous warranty claims. In this type of environment, design flaws and manufacturing problems are worked out over time by changing both products and processes. And in extreme cases, product recalls or retrofits may be needed if the problems are too severe.

Companies that have carefully measured the costs of this approach in wasted production time, proliferation of part types, poor product quality and customer dissatisfaction have concluded that design for manufacturability is an essential part of product design. Many companies are now citing impressive examples of designs with far fewer parts and much simpler manufacturing and assembly requirements than comparable predecessors. The new General Electric dishwashers, the I.B.M. Proprinter, and the Xerox 1065 copier are well-publicized examples of products introduced under this design philosophy.

1.2.2 Quality

Product quality has also returned to a place of pre-eminence in the philosophy of design and manufacturing. Manufacturing strategies no longer rely on traditional accounting views of cost. The hidden costs and poor quality associated with excessive cost competition include inspection time, material loss, rework time, management of irregular-sized lots, warranty costs, and, perhaps most importantly, customer dissatisfaction. In many cases, these hidden costs outweigh the perceived profitability of low cost materials, low cost labor or minimal equipment schemes. In quality discussions, target failure rates of 1 part in a million now replace older targets of 1 part in a hundred. Reaching for the goal of "zero defects" requires steady and continual improvements in process technology; however, substantial gains can also be made by revising operating procedures. Where possible, process control is often far superior
to product inspection: keeping the process within tolerance is better than accurately screening the output. Consequently, part of the thrust for quality improvement has focused on giving operators simple statistical techniques to monitor their processes. This philosophy extends through each stage of manufacturing, even back to inspecting supplier facilities to validate their process control. At each stage, the goal is to eliminate defective materials or products. This system will ultimately reduce cost and improve customer satisfaction.

1.2.3 Just-In-Time

Product flow has been a key area of study in the push to revitalize manufacturing. In many cases, there has been a drive to reduce inventories (raw, in-process, and finished) and to minimize the amount of time material spends in the system. The epitome, variously called "zero inventories," "stockless production," "just-in-time manufacture" or "continuous flow manufacture" demands intermediate inventories be eliminated so that material spends 100% of its time in the manufacturing system in productive, value-adding operations and no time awaiting processing. As a scheduling philosophy, this ideal is a prescription for disaster. Without intermediate inventories, batching and buffer stocks, the output of most manufacturing systems would diminish to a trickle, as naive implementations of the philosophy have shown. However, the ideal focuses attention on how the manufacturing process itself must be improved to eliminate waste in queue time and inventory investment.

In many cases, waste can be reduced or minimized by means of old-fashioned methods engineering (e.g. reducing set-up times through the use of pre-set jigs and fixtures). In some cases, the changes are simply changes in control logic (e.g. increasing the frequency of raw material deliveries and authorizing production runs in small lots on a replenishment or pull basis). In other cases, the changes are more substantial (for example, reorganizing equipment around parts with similar processing requirements and redesigning parts to increase the similarity of processing requirements [group technology]). Sometimes, smooth product flow requires that some equipment be dedicated to certain applications even if that equipment is, in some sense, underutilized. In most cases, the changes are interdependent. For example, in high volume industries (such as the automobile industry), small lot sizes are economically feasible only
if set-up times are short or excess capacity exists. Also, quality is not just a measure of value to the customer; it is a crucial variable which ensures the smooth flow of the production process.

The benefits of smooth product flow go far beyond reduction in inventory investment. For example, long set-up times are often indicative of an inability to guarantee the repeatability of an operation. If an operator must spend several hours tinkering and adjusting a process to bring it within tolerance specifications, then the parts produced on different runs of the process are often noticeably dissimilar. Hence, techniques and investments directed at reducing these set-up times result also in more reliable operations; product quality improves. Also, as production cycle times shrink, the need for long planning horizons also diminish; the scheduling system becomes more responsive to changes in demand forecasts.

1.2.4 Cost Accounting and Engineering Economics

Product flow considerations should be an integral part of the overall competitive strategy. In addition, accounting and economic analysis techniques must be broadened to incorporate these larger issues. Accounting schemes need to extend the range of variables measured (Kaplan [1984]). Economic studies must relate variables in a systems view and derive their implicit costs. Rather than attempting to justify equipment acquisition on a piece-meal basis of direct-labor savings alone, the analyst should begin with the desired performance specifications for the system. Then, investment decisions can be related to those specifications. Hence, underutilization of some equipment may be tolerable if it achieves other goals, such as reduced manufacturing lead times. This approach allows an analyst to measure a strategy's cost and performance and to weigh the options rationally.

1.2.5 Computer Integrated Manufacturing

Another important response to competitive pressure has been the on-going drive to exploit information technology (computers and telecommunications) in all phases of design, manufacture and distribution. Today, computer integrated manufacturing (CIM) normally focuses on the relationship between computer-aided design (CAD) and computer-aided manufacturing (CAM) with issues related to the representation of geometric objects, mechanical analysis of these objects or of the process for
fabricating them, automatic NC code generation and verification, communication protocol for managing information flows among shop floor and control equipment (machine tools, robots, material handling devices, etc.), vision and other sensor systems and so on. While this is only a partial list of CAD/CAM topics under study at universities or in industry, it does suggest that material logistics has been a neglected topic in CAD/CAM activities. Since material requirements planning (MRP) systems are prevalent in industry, many assume that the material flow can be adequately dealt with by tying the MRP system into the real-time control system. Alternatively, specifications for a new planning and scheduling system often require that the system be capable of rescheduling the entire factory immediately. Both assumptions represent unrealistic expectations because they underestimate the difficulty of the scheduling problem.

1.2.6 Scheduling Complexity

The difficulty in solving broad classes of scheduling problems to optimality was one of the chief observations of theorists in combinatorial optimization during the 1970's. Computers can play chess and computers can schedule factories. However, to play well or to schedule well generally requires significant amounts of computer processing time. In worst case analysis, optimization theory now suggests that this time is exponentially long in the problem size. A traditional job shop factory represents a scheduling problem of complexity many orders of magnitude beyond what current computer hardware and software technology can be expected to solve to optimality in a reasonable time frame. Of course, manufacturing managers are willing to settle for less than optimal schedules; but, without careful consideration of the logistics problem, computer integrated manufacturing will run quickly into the limits of technology. Scheduling becomes more important as companies adopt strategies that abandon the commodity aspects of their markets (high volume, standardized products) to foreign competitors and concentrate on high variety, specialty products.

The lesson for those companies opting to employ the CIM approach is that they cannot neglect the focus on product flow. Just as modern design philosophy stresses simplifying product design to the point where a simple robot can perform fabrication or assembly, so the design of manufacturing systems
must reduce product flow complexity to the point where it is realistic to ask a computer to re-optimize the production schedule at a moment's notice. The analogy is striking: companies that have adopted the design for manufacture philosophy have often discovered that human workers in assembly environments become as economic to employ as robots when the designs are simplified; similarly, computers become unnecessary to solve some scheduling and dispatching problems when product flow is streamlined.

1.2.7 Material Logistics and Design

When designing a manufacturing system, material logistics issues must first be addressed in facilities engineering. Given proposed product designs and process flows, the members of the facilities engineering team must establish the numbers and capabilities of each type of equipment, the numbers and types of tooling and fixtures, the physical layout of the facility, the configuration of the material handling system, and the manpower staffing plan. This task requires the engineers to visualize the flow of product through the facility.

Advanced material logistics analysis attempts to understand this flow and its relationship to the physical system. In addition, this analysis allows engineers to plan for the most economical operation of the system. Facilities engineers must therefore work with logistics engineers to establish appropriate lot sizes, production scheduling rules, resource dispatching logic, safety stock levels and storage buffer capacities, so that the facility can achieve desired system throughput rates for all products. The logistics engineers must write the specifications for the information and control systems, and consider equipment changeover or setup times, the repeatability of a process and its corresponding process yield, the reliability and maintainability of process equipment and the complexity of process flow. In the presence of physical and design constraints, such analysis may conclude that large inventories of product or long lead times are unavoidable at certain stages. In such cases, product, process, facility and logistics engineers must work together to understand the tradeoffs, and either devise innovative solutions or agree on the best compromise.
1.2.8 Computer Aided Logistics

Without developing models or conducting extensive computational experimentation, it is extremely difficult to foresee how the interacting elements in these manufacturing systems will affect performance. Therefore, there has been a tremendous effort to complement CIM with software for planning, scheduling, and simulating the operational performance of manufacturing systems. We refer to this area as computer aided logistics (CAL) and assert that CIM should be viewed as a triad of CAD, CAM, and CAL.

1.2.9 Conclusion

This landscape of philosophies and trends provides a perspective from which the economic study of manufacturing and distribution systems can be approached. Our focus is on the importance of material logistics and its centrality in the system design process. Computer aided logistics systems can aid in understanding and improving manufacturing system designs.

1.3 Evolution of Computer Aided Logistics Modeling Systems

A computer aided logistics system is a computer language in which the objects of the language are the objects of material logistics (materials, resources, time, schedules), and the operations of the language are the operations of material logistics (plan, schedule, allocate, dispatch). The mathematical techniques used within the operations of the language --the techniques to evaluate and optimize material flow-- vary widely between systems (discrete event simulation, differential and difference equations, queuing analysis, linear programming, PERT/CPM, etc.). However, the evolution of computer aided logistics systems as representation schemes has followed the evolution of computer languages to higher and higher conceptual levels.

1.3.1 Programming Languages

In attempting to represent the interactions found in a material logistics system, the early designers of computer simulations (e.g. Conway and Maxwell, RAND) made some of the first attempts to provide coherent computer-based representations. These early simulations were written in assembly or
machine language; they provided a way to represent material logistics problems at an extremely low conceptual level. By the early 1960's general purpose programming languages were developed, (e.g. COBOL and FORTRAN) which provided facilities at a higher conceptual level. Rather than work at the level of registers, general purpose programming languages allowed users to work at the level of variables and relieved the user from the extensive bookkeeping of memory allocation. The evolution of general purpose programming languages has continued with the development of higher level procedural languages such as C, ADA and PASCAL, and non-procedural languages such as PROLOG and SQL. These languages are capable of performing complicated operations more easily.

1.3.2 Simulation Languages

General purpose programming languages, like assembly languages, provide sufficient power to describe any manufacturing system, but still remain at too low a conceptual level to be very convenient. General purpose programming languages deal at the level of variables, loops and assignment statements, rather than at the level of events, states, and transitions. As general purpose programming languages have evolved, so have general and special purpose simulation languages, such as GPSS, SIMSCRIPT (Russell, 1976), SIMULA (Dahl and Nygaard, 1966), SLAM (Pritsker, 1984), and SIMAN (Pegden, 1985). In some of these systems, modeling constructs such as resources or carriers are provided. These are conceptual and physical entities commonly found in manufacturing systems, and they offer representation languages at a conceptual level closer to manufacturing systems. With the widespread availability of personal computers and computer graphics, many new simulation systems have been developed (Grant and Weiner, 1986), including XCELL (Conway, et al., 1986), MODELMASTER (General Electric, 1985), SIMFactory (CACI, 1986) and AutoSim (AutoSimulations, 1986). These new systems have taken advantage of new technology by providing visual representations of manufacturing systems.

1.3.3 Optimization Languages

Optimization languages have undergone an evolutionary development similar to general purpose programming and simulation languages. By the late 1950's, the first linear programming codes were developed; a decade later, matrix generators provided representations of mathematical programs at a
level which enhanced the practicality of the technique. Matrix generators share many similarities with programming languages (i.e. looping constructs), and are usually considered to be at a lower conceptual level than the algebraic representations that mathematical programmers commonly use to formulate models. GAMS (Meeraus, 1983) is an example of a newer generation language for describing mathematical programs using an algebraic representation. Structured Modeling (Geoffrion, 1987) uses a different, non-procedural representation to capture mathematical programming problems. GAMS and Structured Modeling represent the current state-of-the-art in general optimization languages.

1.3.4 Planning and Scheduling Languages

Manufacturing planning systems, epitomized by materials requirements planning (MRP) systems (Orlicky, 1955), have undergone an evolution similar to optimization and simulation languages. Orlicky's major contribution was the encoding of the bill-of-materials structure, a representation that greatly structures the planning decision process. As noted by McClain and Thomas (1985) and others, MRP is less of a planning package than a database system that describes the current state of the system, and provides simple predictions of performance. Thus, MRP can be viewed primarily as a useful representation language for manufacturing systems.

Follow-on systems have provided different languages (with accompanying algorithms) to address some of the representational and algorithmic shortcomings perceived in MRP. For example, OPT (Goldratt, 1980) provides a more detailed language than traditional MRP. This allows complicated routings and alternate machines. Furthermore, it provides a sophisticated scheduling algorithm based on the premise that production schedules should be developed around bottleneck machine loading schedules. MIMI (Chesapeake Decision Sciences, 1985, 1986) provides a rich representation language and a collection of planning and scheduling algorithms aimed at improving existing schedules.

1.3.5 Expert Systems

Complex problem representation forms the central focus of expert and knowledge-based systems. They provide convenient languages for representing previously difficult-to-capture constraints and ideas. As applied to CAL modeling, systems such as ISIS (Fox, et al.) are able to capture "fuzzy" or "political"
constraints easily. The power of the expert system languages also provides easy extendability; new
constraints not envisioned by the original designer can be added to the model with relative ease.
Perhaps the greatest shortcoming of the knowledge-based approach is the abandonment of classical
optimization techniques in favor of simple, heuristic search techniques. Although well-suited to arbitrary
knowledge structures, these systems often ignore specialized techniques such as critical path scheduling.

1.3.6 User Interfaces

As computer languages have evolved so has the technology to utilize the languages. Initially,
communication to and from the computer relied on punched cards. In the 1960's, the first interactive
systems allowed individuals to communicate in a somewhat conversational fashion. In those heady days,
people envisioned a burgeoning "Man-computer symbiosis" (Licklider, 1960). By the mid-1970's, fill-in-the-blank or full-screen user interfaces became more common. At that time, the user interacted with a
computer screen as if the screen were a sheet of paper. PILOT (Chesapeake Decision Sciences, 1984)
was one of the first systems that made use of such a user-interface style for production planning.
Although interactive computer graphics was developed in the mid-1960's for computer aided design
applications, its high expense inhibited its wide use until the late 1970's and early 1980's.

One can argue that user interfaces have followed a trend of using familiar paradigms since most
users are already comfortable with the non-computerized version of the paradigm. The "desktop" systems
of the Apple Macintosh and Xerox Star illustrate this design.

Such an interface allows users to draw a representation of a given problem. Shneiderman (1983)
coined the term "Direct Manipulation" to describe this type of interface. As far as we know, the first
system which used interactive computer graphics in the area of material logistics was a job-shop
scheduling system which allowed users to edit a Gantt chart with a light pen (Garman, 1970). Hurrien
and Secker (1976) coined the term Visual Interactive Simulation to describe simulation systems which
allowed users to draw the representation of the manufacturing system in question. The representation
could be animated to show the underlying interactions in the manufacturing system. Bell (1985) has
expanded this concept to Visual Interactive Modeling, and it includes optimization techniques. Several manufacturing simulation systems, including IMMS, XCELL, MODELMASTER, SIMFACTORY and AUTOSIM rely on a direct manipulation interface. GRASS (Jones and Maxwell, 1986) was a manufacturing planning and scheduling system that also relied on a direct manipulation interface. It used an attributed graph to describe the underlying material logistics problem and a Gantt chart to represent a production plan.

Most of the emerging CAL systems, including SLAM, SIMAN, IMMS, GRASS, XCELL and SIMFACTORY, use graphs (networks) as a primary representation format for the underlying model. In general, when using computer graphics to represent logistical models, attributed graphs (Bunke, 1982) are the representation style of choice. However, some users will prefer tabular representations of data over graphical representations (DeSanctis, 1984). Consequently, tabular and text-based representations will not disappear. Nevertheless, graph-based representations are becoming increasingly common. Such representations are not problem-free: large graphs are difficult to draw coherently. Edges cross and the resulting picture looks like spaghetti. One solution to the problem of depicting large graphs is to summarize large graphs in terms of simpler "super-graphs" where one node in a super-graph can represent a whole graph at a deeper level of detail; IMMS uses such a technique. Because attributed graphs have become so common, some have argued for a system to facilitate the development of computer-graphics interfaces for languages that use attributed graphs as a principle representation format (Jones, 1985). The design and implementation of such a system is described in Jones (1985), and Jones and Stine (1986).

On the output side of the user interface, the simplest types of output are descriptive statistics such as means and variances. Detailed lists of the values in the system over time, such as start and finish times of jobs in a production schedule, can also be generated and analyzed later. Given the volume of data that computerized analytical techniques can generate, and given the volume of data needed to describe material flow in a complex facility, computer graphics offers a valuable new technology to summarize and present these data. In the rush to use graphical displays, the importance
of statistics is often overlooked. Nevertheless, the field is still young and many innovations are taking place.

Typical of the graphical displays that CAL systems now provide are plots (over time) of queue lengths and inventory levels. Some systems also provide Gantt charts of machine utilization. Inventory plots and Gantt charts proved their utility long before the advent of interactive computer graphics, but the new technology is adding power to the old forms. For example, GRASS enables the user to make changes to a Gantt chart production schedule and to see the impact of these changes on inventory plots in real time. Inventory plots and Gantt charts can even be animated to some extent by drawing them while the simulation runs. More complex animations of the dynamics of the manufacturing system are also provided by such systems as TESS (Standridge, 1985), CINEMA (Pegden, et al., 1985), XCELL (Conway, et al., 1986), MODELMASTER (General Electric, 1985) and IMMS (Engelke, et al., 1985). Such systems typically change the appearance and location of icons over time. Such icons could represent machines, material and carts. Animation is most useful during the development and validation phase of a simulation model. When the user is ready to perform experiments on the model, statistical analysis appears to be the preferred technique.

1.3.7 Framework for a CAL System

Computer aided logistics differ in their purposes, their representation style and their analytical techniques. The following framework is useful for designing and investigating a CAL system. The presentation of the COSMOS system in the next section is organized around this framework.

A. **Representation:** What are the kinds of material logistics problems of interest? How are these problems represented? What is the scope of decision models that can be supported from the same data representation?

B. **Optimization:** What are the material flow decisions that must be made? What formal modeling or analytical techniques are appropriate for studying these decisions?

C. **Evaluation:** What attributes of system performance are of critical interest, how can they be measured, and what techniques will provide useful estimates of these attributes?
D. **Integration:** How are the elements of the representation schemes and optimization procedures linked and applied in understanding material flow behavior in particular manufacturing environments?

1.3.8 Summary

A CAL system is not simply a simulation or optimization model, nor is it merely an expressive language for representing problems to be solved by a single analytical paradigm. A CAL system has (or should have) a language for representing the system of interest irrespective of the eventual analysis techniques. A CAL system should concentrate as much energy on delivering the technology as on the underlying analysis techniques. As such, it must integrate the theoretical and empirical results not only from the fields of operations research, operations management, management science, and industrial engineering, but the fields of computer science, artificial intelligence, user interface design, and database management. In the next section, we present an overview of our implementation of this integration—COSMOS.

1.4 **The Cosmos Modeling System**

1.4.1 Representation

Figure 1 shows the structure of COSMOS. It is organized into functions associated with the four components of the design framework for a CAL system presented in Section 3. Representation of the facilities, the products, and the processes used to manufacture products and their components is the backbone of the COSMOS system. The database represents these components through three fundamental graphical charts: the Resource Chart, the Product Structure Diagram, and the Operation Task Chart. In essence, these three charts contain all the information about what resources are available, what products or components are to be made, how the products are to be made (including the resources required to perform each operation for each component), product demand or delivery schedules over time, raw material or component delivery schedules if appropriate, process yields, maintenance requirements and
FIGURE 1. COSMOS VIEW OF CIM:

- **Design**
  - Facilities
  - Products
  - Processes

- **Database**
  - Resource Chart
  - Product Structure
  - Operation Task Charts

- **Planning System**
  - Demand Forecasting
  - Production Planning
  - Resource Scheduling

- **Schedules**
  - Customer Shipping
  - Production
  - Supplier Delivery

- **Control System**
  - Resource Dispatching

- **Performance Measurement**
  - Resource Utilization
  - Inventory
  - Throughput and Flowtime
  - Financial/Economic Measures
many other key data elements. In summary, the database represented by these charts captures the physical elements of the represented environment. We now illustrate these three charts through an example.

1.4.2 An Example

Suppose a manufacturer of pneumatic tools is contemplating the purchase of new machine tools to streamline production. These machine tools, among other things, offer the manufacturer the opportunity to reduce work in process (WIP), manufacturing cycle times, and space requirements. A group technology approach is planned which should improve material flow. Thus, material logistics issues are central to the decision to change the manufacturing environment. Our example is based on a simplification of this situation.

In our simplified view of the manufacturing environment, let us assume that one of the cells in the new facility is designed to contain equipment to machine and finish two component types. The output of this cell is combined with components that will be made in other portions of the factory or that are purchased from elsewhere. The two components are called a handle and a housing. Each arrives to the manufacturing cell as a new casting. Complicated machining operations, such as milling and drilling are accomplished on a single vertical milling machine. The setup and machining time needed to meet demand for these two components is sufficient to consume almost all the available time on one such machine. Since there is sufficient capacity and the cost is substantial, there is only one mill in the proposed cell. After a batch of a housings or handles is completed on this milling machine, some finishing operations are also performed in the cell. Since the cost of this equipment is low, dedicated finishing equipment exists for the housings and handles. Due to the processing time and the desire to minimize WIP, there are two finishing machines for housings. There is only one finishing machine for handles. Once they complete the machining and finishing operations, handles and most housings are delivered to a nearby assembly area. Some of the housings are produced to meet demand for spare parts and they are packed and shipped to customers according to a spare parts demand schedule. The remaining housings are combined with the handles and other parts and are assembled into finished tools.

1-15
Part of this assembly process is a performance check. This assembly and test step is conducted at a new automated work station that works on one tool at a time. If a tool passes the test, it is placed in finished goods stock. The few tools that do not pass the inspection test are reworked at a rework station within the assembly department. Sometimes it is determined that the tool cannot be repaired and it is scrapped.

In this example, the elements we have chosen to represent with the COSMOS modeling system are the machining, finishing and assembly operations and the corresponding flow of material. The representation of the database for this problem begins with a description of the resources. The Resource Chart allows the modeler to represent various categories of resources (machines, labor, tooling, fixtures, storage, track, carriers, stations) and specific types and instances of these resources. In our example, we have used two categories of resources: machines and storage. For example, we have assumed that labor and material handling are not constraining factors. Given these two categories of resources, we then identify the resource types within each category.

We have specified five resource types in the machine category: milling, finish (handle), finishing (housing), assembly station, and rework station. Except for the housing finishing equipment, there is one instance of each of these resource types. There are two instances of the finishing machines for housings because of the workload as discussed previously. There are six storage areas, each containing inventory that exists in various stages of the production process.

Observe that we have not considered material as a resource. Rather material is modeled separately, since its movement will be governed by material logistics policy and physical resource constraints. The other resources are all assumed to facilitate the flow of material. The Resource Chart, like all COSMOS charts, is constructed using an interactive, menu-driven, computer graphics editing system. In describing these charts in later chapters, we will be using the language of computer science ("graphs of nodes and edges") rather than the older language of operations research ("networks of nodes and arcs"). The Resource Chart for our example is displayed in Figure 2. This graph is used to indicate the logical organization of resources rather than their physical location.

1-16
FIGURE 2.
Cosmos Version 2.0: RESOURCE CHART Graphical Editor

editor: THORNMK4 RSC

pan left
pan right
pan up
pan down
zoom in
zoom out
reset

view node
edit node
move node
delete node
add node

edit rtyp

generate
saveename
file
help
exit

locator input requested
FIGURE 3.
Cosmos Version 2.0: PSD Graphical Editor

editor: THORNMK4 PSD

locator input requested
1.4.3 The Product Structure Diagram

Once the resources have been identified, we next construct the Product Structure Diagram. This diagram documents the routings and bill-of-material relationships that exist. The diagram corresponding to our example problem is displayed in Figure 3.

The Product Structure Diagram is a directed acyclic graph, whose nodes represent either an operation of some type or a part number. The edges represent flow of material from one operation or part number to its successor.

There are four different icons used for nodes in this example. They represent receiving ( ), shipping ( ), a manufacturing process step ( ), or a part number ( ). As in the Resource Chart, different colors are used to distinguish different node types. We will now discuss each of these node types.

There are three receiving nodes in the diagram representing the receipt of raw handle castings, raw housing castings and other parts. Associated with each such node are data reflecting the schedules of arriving material. For example, these data stipulate the models for the timing and quantity of material receipts; both timing and quantity can be random variables.

Once each type of material has been received or completes a process operation, it is assigned a corresponding part name. The node representing the part is either a terminal node in the graph, or it represents material required to complete a subsequent operation (e.g. a further process step or a shipping operation). Using an Edit Node command, we can set values of the control parameters for each part name. This allows us to establish when that part should be replenished. Kanban, (s,S), reorder interval or other trigger rules can be specified by entering appropriate data. Scheduling mechanisms other than replenishment strategies will be discussed later.

The process node represents a transformation of a part into a subsequent part. Note that part names do not necessarily correspond to part numbers found in an accounting system. For example, we identified two part names with unfinished handles in the graph (HANDLE_CASTING and
HANDLE_PRE_FINISH) where there would probably be only one part number in an accounting system. We have done this so that we can identify and track WIP.

At this point, we are not interested in the details of the physical aspects of a process. That is, we have named a process HNDLE:MILL/DRILL, but have not established parameters for the process (e.g. the resources needed to accomplish this operation, how long the operation will take for either setup or run, or the possible rework or scrap implications). Thus, a process operation node is only used to represent a transformation of one part name into another. While the names on the process nodes in the example represent some physical change in the shape or geometry of a part, process operations in other settings could reflect other events. For example, the movement of inventory from one warehouse to another in a distribution system or the flow of material from one tank to another in a process-oriented industry could be modeled by this type of node.

There are two shipping nodes in the example. They are used to establish exogenous consumption patterns for various material. In our example, the ASSEMBLY_DEMAND node has data associated with it that reflect the timing and quantity of customer demand. The SPARES_DEMAND shipping node has similar data associated with it. These times and quantities can always be specified as random variables.

1.4.4 The Operation Task Chart

The final chart type in COSMOS is called the Operation Task Chart. Each operation found in the Product Structure Diagram has an associated Operation Task Chart. The Operation Task Chart is also a directed acyclic graph. The nodes are used to indicate what resources and material are needed to carry out each step of the operation, the times required to complete these steps, storage capacity needed, process batch sizes, the sequence in which they must be conducted, etc. There are three edge types, which primarily represent precedence, the initiation sequence of tasks (push or pull flow) and the probabilistic outcome of inspection tasks. There are six task types that can be used to model various aspects of an operation: draw inventory ( ), store inventory ( ), setup a machine ( ), run a machine ( ), move a resource ( ) and inspect material ( ). Each node and edge type also has an associated color.
A user could develop an Operation Task Chart for the ASSEMBLY operation like Figure 4. It illustrates the use of some of the node and edge types discussed in greater detail in subsequent chapters. The initiating task for the assembly of a finished good is represented by the ASSEMBLY_STATION node. To begin the assembly task, the needed amounts of handles, housings, and other parts are drawn from the appropriate storage areas. After each unit is assembled, it is inspected at the ASSEMBLY_STATION. If the unit passes the test, it is stored as a FINISHED_GOOD in the designated storage area resource. If it fails the test, rework at the REWORK_STATION is performed. If the unit cannot be repaired, it is scrapped; otherwise, it is sent to the FINISHED_GOODS storage area.

The edges emanating from each inspection type node carry information on the probability the tested part will follow that path. For example, the edge connecting the ASSEMBLY_STATION and FINISHED GOODS has an associated probability of 0.9. Thus, each completed assembly has a 0.9 probability of being accepted immediately and a 0.1 probability of requiring rework.

Very complex Product Structure Diagrams and associated Operation Task Charts can be constructed to represent a variety of situations. The versatility of the COSMOS modeling language can only be appreciated by actually constructing models for a broad range of manufacturing environments. This brief example illustrates only a small portion of the system’s power.

1.4.5 Optimization: Planning, Scheduling and Dispatching

Once the environment has been represented, there is an opportunity to apply analytical techniques to plan future activities. As stated in Figure 1, this could include forecasting future demand, planning production for components and assembly of finished products, and scheduling resource availability and usage. COSMOS has been designed to allow the user to access flat files containing the appropriate elements of this representational database. These data can then be used to generate optimization models or appropriate heuristics. For example, it is possible to study the impact that various planning rules or algorithms will have on system performance. Of course, this is a key reason for developing this modeling environment. However, in a practical sense, it is necessary to employ procedures that quickly generate plans and schedules that use resources effectively and that meet production targets on time.
locator input requested
The nature of the planning process and the use of models within that process have been described by many authors. See Hax and Meal (1975) or Maxwell, Muckstadt, Thomas and VanderEecken (1983) for a discussion of theoretical modeling and references to the discussion literature on this subject. It is not our intent to make the case for modeling and decision tools. Rather we will now illustrate, using the example problem, how modeling tools are an essential part of the COSMOS environment.

1.4.6 Cyclic Scheduling

In the example, the representation of the physical environment (resources, products and processes) and the customer demand schedules were established in the Resource Charts, Product Structure Diagram and Operations Task Charts. Given these data, we would like to determine a production plan and schedule for the manufacturing cell which is consistent with the resource constraints and the customer demand schedule. There are many ways to generate these plans and schedules. However, for the environment that we have described, a method proposed by Jackson, Maxwell and Muckstadt (1984) is particularly suitable. We recommend that the reader study this paper to obtain the details of a proposed scheduling system. The method is based on the idea that, in situations characterized by relatively stable product demand, it is desirable to produce each part or to assemble the final good on a regular cycle. That is, the time between the initiation of successive production runs for each part or the assembly of a batch of a finished good should be approximately the same. The exact frequency depends on the value (cost) of each raw material, the value added at each operation, the time available for setup and run on each machine, the setup, run and assembly times for each machine and the cost of holding inventory. A base planning period of a week was also set for this problem; that is, successive production batches will be at least one week apart. We also assume that the production batches will be sized to be an integer multiple of this base planning period.

Given these restrictions, we decided that the first operation for both housings and handles should be started once every two weeks. This result occurred because the milling machine was very highly used. It could not accommodate more frequent production, since there was a rather high setup time for each part type. The remaining manufacturing operations occur as follows. Assembly occurs on
essentially a continuous basis. The finishing operations are begun as soon as parts become available from the preceding process.

Once this production frequency has been fixed, we then establish a cyclic production schedule. That is, we determine the timing of the starts of each batch on each machine so that the pattern of production repeats indefinitely. The approximate Gantt chart corresponding to the solution is given in Figure 5. Note that the time between initiation of production batches is the same, although the exact timing of the beginning of a batch and duration of the run length do not necessarily correspond to the start of a week.

1.4.7 Safety Stock

Recall that not all assembled units pass inspection. Hence, to achieve the desired shipping schedule, it is necessary to determine the amount of buffer (or safety stock) required to meet this schedule. For this example, a small amount of finished goods buffer stock and a small amount of raw material and pre-finished handles and housings safety stock are desirable. The exact amounts depend on the chosen level of service for the finished good. We chose these levels to enable us to meet all customer demand on time with a probability greater than 0.99.

There are a number of interesting observations to be made when establishing production schedules and safety stock levels. For example, the safety stock levels we selected depended on the size of the transfer batch from operation to operation. Here, the transfer batch is the number of units that are required to be in a material handling container at the time the container is moved from one machine to the next. We have observed that the larger the transfer batch size, the more safety stock that is needed; that is, the selected capacity of the material handling containers influences the needed amount of safety stock.

The production schedules and safety stock requirements we have established yield minimum WIP. If the environment were changed, these schedules and safety stock levels would also change. For example, by lowering setup times on the milling machine, the frequency of batch starts for handle and housing castings could increase. This, in turn, would reduce both cycle WIP and safety stocks. Thus,
FIGURE 5.
Cosmos Version 2.0: Resource Utilization Plot

resource utilization plot: BILL

ASSEMBLY1
FINISH1
FINISHING1
FINISHING2
MILLING1
REWORK1

TIME

locator input requested

new plot
pan left
pan right
zoom in
zoom out
page up
page down
reset
report

help
exit
by using the COSMOS modeling system, one can evaluate how changes in product and process design affect the schedules produced by an optimization algorithm or heuristic. This evaluation can also help measure the impact on material flow and customer service.

1.4.8 Evaluation

The effect of employing production scheduling rules suggested by an optimization procedure or heuristic is not easy to determine in advance. Generally, the underlying mathematical models are crude approximations to reality. In many models, equipment is assumed perfectly reliable, setup and run times per piece or batch are fixed and known, all resources are available in adequate amounts, raw materials are delivered on time in the proper amounts and are of perfect quality, lead times are known and fixed, waiting times are specified and so on. Even when probabilistic models are used, they focus on relatively small portions of a large problem due to computational and algorithmic complexity. Thus, to evaluate how well rules will perform in a particular manufacturing environment requires more than estimates obtained from a simplified mathematical analysis (such as those produced from MRP systems, for example).

Measuring the interaction between planning and scheduling rules and the physical environment normally requires the use of simulation. Simulations can be constructed at varying levels of detail to study different types of questions. As is the case with optimization procedures, the modeler must precisely describe what questions should be studied and the appropriate level of modeling complexity. Clearly, if the simulation attempts to capture all aspects of reality, it will be virtually impossible to validate a model much less run the simulation and try to identify cause and effect relationships. While the benefits of using simulations are well known, so are the pitfalls. Thus, we remind the COSMOS user to relate the needed information to the model's size and complexity.

COSMOS contains a very powerful simulation capability. After answering a few simple questions (e.g., How many time units do you want the simulation to run?), the user can initiate a simulation run to evaluate the performance of a material logistics system that has been modeled using the previously mentioned charts. Once a run is completed, the user can view the results of the simulation using a
menu-driven windowing system. In addition, the user can simultaneously view the results of many simulation runs and each run can be evaluated in several different ways.

There are 6 types of charts and plots that can be displayed; each portrays a different aspect of the system’s performance. In combination, they allow a thorough understanding of the system’s behavior. These output displays are called the Inventory Plot, Net Inventory Plot, Backlog Plot, Shortfall Plot, Material Cycle Time Plot, and Resource Utilization Chart. In addition, various other textual and graphical information gives statistical measures associated with inventory levels and backlogs for each part number. The details of the commands needed to invoke these displays are discussed in the following chapters.

1.4.9 The Inventory Plot

The Inventory Plot indicates the amount of on-hand stock at each point in time over the length of the simulation horizon. The user can specify one or more part numbers to display at a given time. The part numbers are differentiated by their color on the plot. Figure 6 is an example of an Inventory Plot. Inventory for only one part type, finished goods, is graphed. The abscissa represents time and the ordinate represents units of inventory on-hand. The graph then shows how on-hand stock of the finished good varies over time as a function of the inventory and scheduling policies that have been selected, the system’s resources, each component’s routing, set-up and processing times, and reject and repair rates. The Net Inventory Plot is similar to the Inventory Plot; the only difference is that net inventory (on-hand minus backlog) is plotted instead of on-hand inventory. Figure 7 shows how net inventory (on-hand minus backorders) for three part types--handle castings, housing castings and finished goods--change over time. As shown in the graph, a request for about 100 handle castings on days 9, 19 and 29 cannot not be satisfied immediately; this results in backorders for this part. However, observe that these shortages do not affect the ability to meet the delivery schedule for the finished good on time. That is, the graph shows that there are never shortages for the finished good.
FIGURE 6.
Cosmos Version 2.0: Inventory Plot

inventory plot: THORNMK

FINISHED_GOOD

locator input requested

inventory
mat cy lo
mat cy up
net inven
back log
shortfall
pan left
pan right
zoom in
zoom out
reset

help
exit
FIGURE 7.
Cosmos Version 2.0: Inventory Plot

inventory
mat cy lo
mat cy up
net inven
back log
shortfall
pan left
pan right
zoom in
zoom out
reset

locator input requested
1.4.10 The Resource Utilization Chart

The Resource Utilization Chart displays the usage of each resource type over time. For each non-material resource type, this chart shows the percentage used over the time horizon. In the example problem, each resource can process only one part at a time. Thus, the utilization for each machine is either 0 or 100 percent at any particular moment. Figure 5 shows the Utilization Chart for the example. Recall that the milling machine (Milling 1) is used to process both housings and handles. The cyclic schedule requires handle castings and housing castings to be processed every second week. During the first five day period (one week), the milling machine works only on handle castings. The graph shows that the mill is first appropriately set up and then it processes a batch of handle castings. Color is used on the screen to indicate this setup-process sequence. Beginning on day 6, the housing castings are processed on the mill. This same processing cycle repeats on the mill during the periods of time 10 through 20 and 20 through 30. The machine types Assembly Station, Finish1, Finishing1 and Finishing2 all have high utilization rates. During each period when a machine is busy, the graph sometimes indicates that one process batch completes and another one begins immediately. For example, the first two process batches on the Assembly Station occur continuously.

1.4.11 The Backlog Plot

The Backlog Plot is used to display how closely the actual shipments of a part number match the desired shipping schedule. Recall that for each shipping node in the Product Structure Diagram, there corresponds a shipping schedule. This schedule determines the timing and quantity of desired shipments of the corresponding part number. The Backlog Plot is a plot where the abscissa measures time and the ordinate measures inventory. There are two graphs displayed for a given part number. The first measures cumulative actual units shipped and the second measures cumulative units to be shipped according to the actual shipping schedule. At a given point in time, the first graph either coincides with the second, or it indicates that cumulative shipments are less than required through that time. Thus, the first graph is always on or below the second; the difference between them measures the backlog at a point in time. The actual schedule can reflect the randomness of the customer demand
process. Thus, for example, although we might expect to have 100 units demanded per week, we could actually have 115 units demanded.

1.4.12 The Shortfall Plot

The Shortfall Plot is similar to the Backlog Plot. The only difference is that rather than plotting an actual shipping schedule, as is done on the Backlog Plot, the planned or expected shipping schedule is graphed. Thus, there are two graphs displayed for a given part number, the first showing the actual shipments and the second showing the expected shipping schedule. By viewing both the Backlog and the Shortfall Plots, it is possible to identify whether shortages are due to resources, policy decisions or to randomness in the demand process.

1.4.13 The Material Cycle Time Plot

The purpose of the Material Cycle Time Plot is to display the movement of a given material type through the material logistics system. Again, the abscissa represents time, and the ordinate depicts inventory. Figure 8 illustrates this plot. The user first picks a part number, typically that of a raw material. We have chosen the part handle casting. Recall that the Product Structure Diagram indicates the bill-of-material relationships for all parts. The Product Structure Diagram shows into which part numbers the handle casting goes and in what quantities. The handle casting is first transformed into the part called handle prefinish, then to the part named handle, and finally to the part named finished good, or, in the case of rejects, scrap. These different stages of transformation are referred to as levels; level 0 corresponds to the handle casting and level 3 corresponds to the results of the assembly operation, finished good and scrap. For each level a graph is constructed showing cumulative receipts of production of all part numbers at this level, measured in the units of raw material selected. If there is inventory present in the system at the beginning of the simulation run, the graph for each level is shifted upwards by the total amount of inventory at that level or at levels below (measured in the units of the raw material). The graphs for each level are displayed on a single plot. Thus, the uppermost graph indicates the cumulative available inventory of the part selected by the user (in this case, the handle casting). The next graph corresponds to the usage of the selected part (the handle prefinish) in
FIGURE 8.
Cosmos Version 2.0: Inventory Plot

mat cy lo: THORMK  #HANDLE CASTING

inventory
mat cy lo
mat cy up
net inven
back log
shortfall
pan left
pan right
zoom in
zoom out
reset

locator input requested

help
exit
all immediate successor operations indicated in the Product Structure Diagram. Note that this second
graph must be on or below the graph of the one for the handle casting since cumulative production of
successor parts cannot exceed cumulative production of predecessor parts. Each subsequent graph on the
plot has a similar relationship between predecessor and successor levels in the Product Structure
Diagram. There are four graphs displayed in the plot indicating in order the cumulative production or
material receipt of handle castings (upper most graph), pre-finished handles, handles, finished goods, and
scrap. In addition, a fifth graph is constructed showing cumulative shipment or scrapping of all part
numbers that have no successor levels. Hence, the lowest graph in Figure 8 represents cumulative
shipment of finished goods plus cumulative scrap.

The Material Cycle Time Plot presents a great deal of useful information in graphical form. For
example, the slope of the lowest graph, the "ship plot," represents the throughput of the machine cell
measured in terms of the raw material, handle casting. Suppose a particular time is selected. By
measuring the difference between successive graphs, we can determine the amount of the original part
that is in each stage of production or in a finished product at that time. For example, look at time 10
on the graph. The total number of handle castings received by time 10 is 285, including 85 units on
hand at time 0. The number of pre-finished handle castings completed by this time is 175. Thus at
time 10 there are 110 handle castings awaiting processing. Observe that 110 units of the finished good
were shipped by time 10. Therefore there are 175 handle castings in various portions of the production
process at time 10. (The exact inventory levels can be found by locating the cursor on a particular
point on the graph; the time and inventory levels are then automatically displayed to the user.) On the
other hand, if a particular inventory level is selected on the ordinate, it is possible to track the entire
history of the specific instance of the part number through the system. The arrival time of the part is
indicated by the first graph. It is the time at which the cumulative stock of the part number reaches
or exceeds the given value. For example, consider the arrival time of the 285th handle casting, which is
time 10. The length of time that the part stays in its original state is equal to the difference in time
between the times the first and second graphs reach or exceed the chosen inventory level. Thus the
285th handle casting waits until time 23.3 before it is processed. By continuing in this manner, the user can measure the length of time each part remains in each state and ultimately can measure the total time the part remains in the system. The 285th handle casting is shipped in a finished good at time 27. It remained in the system for 17 time units. This is only an approximate calculation because the simulator is tracking batches of inventory rather than individual pieces. Nevertheless, it is possible to calculate the total inventory of a part number that is carried throughout any portion of the time horizon and to determine the distribution of the time that the parts remain in the system. It is well known that the average inventory level must equal the average time a part remains in the system times the actual output shipping rate of the final products in which the part is used. The purpose of the Material Cycle Time Plot is to convey this information of throughput, inventory investment and material cycle time in a graphical fashion.

By viewing various inventory, net inventory, utilization, backorder, shortfall and material cycle time plots, it is possible to understand the interactions that exist among scheduling rules, resource availability and safety or buffer stock levels. This gives the user the ability to truly diagnose the system’s dynamic behavior.

Each of the above plots and charts can be viewed individually or in combination. The COSMOS windowing system allows viewing of a product structure diagram, task chart and as many different plots as desired at one time. More than one case can be viewed at a time as well, thereby permitting comparisons of a system’s behavior to be made when resource levels, scheduling rules or other data elements are revised. Furthermore, zoom and pan features exist that allow the user to examine in detail any interval of time in the simulated horizon.

1.4.14 Integration

Integration primarily refers to the research activity of the COSMOS development project. COSMOS is structured in such a way as to permit a variety of planning, scheduling, and dispatching regimes to be installed. The simulator is driven by means of production orders called lot releases and dispatching decisions within the simulator are made based on lot release priorities. How these lot
releases arise and how these priorities are set are the questions addressed by the planning, scheduling and control system. Different industrial environments call for different approaches to the material logistics problem. COSMOS is intended to support many such material logistics systems, including those developed by the user. The nature of the support lies in the open structure of the database and the simulator. The database is stored in flat files that can be accessed and modified by outside planning and scheduling software. The simulator has well defined decision points at which dummy procedures (code that performs no action) are called. Users familiar with the PASCAL programming language can replace these procedures with dispatching procedures that make scheduling and dispatching decisions according to the users' special rules. Users without the resources to develop their own planning, scheduling, and dispatching software will be able to choose from libraries of systems that we or others develop for inclusion in COSMOS. At the present time, this library consists of replenishment style production control strategies. Other strategies are under development and are the subject of a major research effort.

1.5 CONCLUSION

COSMOS is a modeling environment that truly allows the user to represent, optimize, evaluate and integrate information related to the design and operation of a manufacturing system. Because it is an object oriented modeling language using a computer graphics interface, it permits manufacturing engineers unfamiliar with computer programming to develop and test extensive models of manufacturing operations. By its structure, COSMOS leads the engineer to consider issues of planning, scheduling, safety stock, dispatching, and information flow in the course of studying the impact of different physical configurations of equipment and material flow. It is our hope that manufacturing engineers who have conducted modeling studies using COSMOS will be better prepared to tackle the broader issues of computer integrated manufacturing because they have developed an understanding of material logistics fundamentals.
Chapter 2

BASIC CONCEPTS IN COSMOS

COSMOS has a rich language all its own; the terms and concepts in COSMOS form an amalgam of the languages of Computer Science, Operations Research and Industrial Engineering. To exploit all the features of COSMOS, the user should become comfortable with the ideas and vocabulary in bold face. The descriptions of the COSMOS editors, menus and screens will make frequent use of this terminology.

2.1 Tasks

The fundamental entity in the COSMOS modelling system is the task, an activity that consumes time and resources. A task can be as general as "assemble a satellite" or as detailed as "grasp tool with right hand." As in all modelling exercises, the level of detail should be determined by the nature of the issues being addressed and the accuracy needed in the answers. Once initiated, simulation studies have voracious appetites for data and computer time; it is wise to set their diets before they are released.

The description of a task is captured in a task node; the attributes of a task node include a specification of the random variable that determines task time.

2.2 Task charts vs. Precedence Diagrams

Task nodes are elements of a larger structure: a graph of nodes and edges describing a sequence of tasks called a task chart. To illustrate the definition of a task chart, we contrast it to the notion of a precedence diagram. In a precedence diagram, an edge from node A to node B means only that the task associated with node A must be completed before the task for node B can begin. The choice of the actual timing of tasks in the simulation would be left to the dispatching logic within the simulator or to an intelligent simulation model preprocessor. This logic or preprocessor would need to anticipate completion times of sequences of tasks and initiate subsidiary tasks accordingly. Different objectives ("finish this job first" or "keep the bottleneck machine busy") would result in different task initiation
sequences. This is the approach suggested by Chatterjee et. al. in modelling manufacturing systems. Though ultimately desirable in the flexibility it offers, such an approach is more fundamental than the one implemented in COSMOS. In a task chart, the sequence of tasks is completely specified. Edges in the graph have the interpretation of task triggers; that is, the initiation or completion of one task triggers the initiation of another task. The burden of determining a good task sequence is left to the modeller.

2.3 Task Charts

Within the framework of a task chart, however, the modeller still has considerable flexibility. There are three types of edges or triggers that can be used to describe a sequence: push, pull, and probabilistic. A push edge from node A to node B means that when the task at node A completes, the task at node B should be initiated. This is the most common type of edge. A pull edge from node A to node B means that if node B is triggered (presumably by a push edge that enters B or by some other mechanism), then the task at node A should be triggered as well, and the task at node B should not proceed until the task at A is completed. This is useful if the primary task being initiated is, for example, an assembly step and this step should, in turn, trigger task sequences to fetch the necessary components and wait until all these sequences complete before proceeding with the assembly. Some task sequences, such as scrapping or reworking a batch of parts due to process failure, are probabilistic in nature. These can be modelled using a probabilistic edge. Probabilistic edges have the same interpretation as push edges except that one (and only one) probabilistic edge emanating from a node is used to trigger a succeeding task. (Inspect nodes are an exception to this rule. They are described below.) Each probabilistic edge has a probability weight attached to it as an attribute, and an edge is selected randomly according to these weights.
2.4 **Precedence Edges**

The trigger edges --push, pull and probabilistic-- define task sequences. A precedence edge from node A to node B simply means that the task at node A must be finished before the task at node B can begin. No triggering of A by B or B by A is implied. A and B must be triggered independently via lot releases or trigger edges; the precedence edge merely defines a timing constraint. In most situations in which a precedence edge appears necessary, the modeller could accomplish the same effect with pull edges and some modelling tricks such as the triggering of dummy nodes. Therefore, precedence edges are a redundant feature; they are included in the modelling language to allow a more natural treatment of these situations.

2.5 **Infinite Loops and Root Nodes**

So far, we have described how tasks at one task node can trigger tasks at other task nodes along push, pull, or probabilistic edges. If the task chart contains a loop of edges, and if any one of the task nodes along the loop is triggered, it is possible for the sequence of tasks generated to grow without limit. The tracking needed for such growth could consume all available computer memory. Thus, when viewed as a graph of nodes and directed edges (with the orientation of the pull edges reversed), the task chart must be acyclic. Nodes which cannot be triggered by any other node (i.e. nodes that have no incoming push or probabilistic edges and no outgoing pull edges) are referred to as root nodes. Root nodes are triggered by a separate mechanism called lot releases, described below. Lot releases are central to production control, but before addressing them, it is important to describe the relation of material and other resources to tasks.

2.6 **Material Flow**

Material is a malleable concept in manufacturing. Sheet metal is cut, pressed, shaped, welded, sanded and painted. Crude oil is cracked into a hundred different distillates. Plastic pellets are molded into a bewildering variety of shapes. The addition of one circuit board to a dishwasher changes its model
number. Screws are packaged in bags, bags in boxes, boxes in containers. The units of material measure, what is meant by "material," can change at each step in the production process. Accordingly, material in COSMOS is a local concept, associated with the task. Tasks operate on material, the units of which are an attribute of the task node. In a cutting step, for example, the material would likely be measured in terms of units of the input material (e.g. sheets of steel), and the time to complete the task would be expressed as so many minutes per sheet. A subsequent step would operate on the cut pieces, its time to complete measured in minutes per piece. In an assembly step, the material would likely be the output, the assembled product and the task time expressed as so many minutes per product. Conversion between units of material measure is represented along the edges of the task chart. Each edge has, as an attribute, a transfer coefficient that relates the units of material measure in the "from node" to the units of measure in the "to node." Hence, a cutting step that cuts four identical pieces from one sheet of metal would trigger the subsequent step with a transfer coefficient of four. For an assembly step, the units would typically be the units of assembled product. And, the transfer coefficients on incoming edges would effectively describe the inverse of bill of materials quantities of the assembly components. Hence, if the assembled product requires twenty units of a particular component, the transfer coefficient from a node denominated in the units of the component to the assembly node would be 0.05.

2.7 Transfer Batches

The continuum of material is captured by the notion of transfer batches. The simulator views material as coming into being at the beginning of a task and disappearing upon its completion. The flow of material is a simulated illusion as one task triggers another, and the triggered task inherits the quantity (modified by the transfer coefficient) of the triggering task. However, without further accounting, the illusion would fail when there are lapses of time between when the triggering task completes and the triggered task can begin. For example, the triggered task may trigger other tasks along pull edges and wait for their completion. The logic of the simulator is such that the material associated with the
triggered task is not accounted for until the task actually begins. To represent this inter-task material, the simulator creates a temporary record called a transfer batch whenever a task completes. The transfer batch measures the amount of material (in units of the now-completed task) which is awaiting processing by subsequent tasks. As the tasks triggered by the completed task begin (there can be many such tasks), the transfer batch is reduced in size according to the size of the triggered tasks and the transfer coefficients on the edges that triggered them. When all the triggered tasks have begun, the size of the transfer batch must be zero, and the simulator discards the record. The simulator creates these transfer batch records only if the task node has push or probabilistic edges emanating from it. Hence, in the simulated flow of events, material will disappear if a task node is terminal (i.e. if the task node has no such emanating edges). Continuity of material in this case can be captured by assigning a part number to the material and by making the terminal node a store node (discussed below).

2.8 Batching

A problem arises in modeling the batching of material. For example, a task such as an assembly step may be triggered with a large quantity (i.e. an order to assemble a thousand units). Are subsequent steps to wait for all one thousand units to complete assembly, or can they begin as soon as the first units come off the assembly line? To allow for both possibilities, a transfer batch size attribute is associated with each task node. If a task node is triggered with a quantity larger than the transfer batch size, then the quantity is broken up into smaller batches, each no larger than the transfer batch size, and a task is initiated for each batch. The time to complete a batch would be proportional to the number of units in the batch. The first batch to complete then triggers subsequent task nodes along push edges, passing along the batch quantity adjusted by the appropriate transfer coefficients. Each successive batch to complete follows the same process, triggering subsequent tasks with the appropriate batch quantity. Thus, the task chart is a model for the dynamic creation of tasks; each task node can give rise to many task instances in the course of a simulation. Since the simulator keeps track of each task that is triggered, it can quickly use up all computer memory allocated to it if the modeller has
specified very small batch sizes for very high volume tasks. Here again, obsession with model completeness ("the model isn't realistic unless it tracks every nut and bolt") must be tempered with practicality. Some tasks, such as product curing, heat treatment, washing, or equipment cleanup, are not associated with material or are not time-dependant on the number of units in the batch. To accommodate these cases, the task time in the task node can be specified as either time per batch or time per unit. If the per unit switch has been selected, then the task completion time is generated as the sum of \( n \) independent random variables (as described in the task node), where \( n \) is the number of units in the batch. For certain probability models (constant, Poisson, normal) a single random variable is generated, with mean and variance scaled by \( n \). Otherwise, if the per batch switch is selected, the time to complete is generated strictly according to the random variable description in the task node; the number of units in the batch is ignored. Thus, a task can be initiated with a quantity of zero and still be meaningful to the simulation.

2.9 Information Flow

This latter point deserves emphasis because there is no requirement that a sequence of tasks have any material flow associated with it. Tasks can be triggered with zero quantities and these tasks can trigger other tasks again with zero quantity. Also, even if tasks are triggered with quantities, there is no requirement that the quantities being passed from task to task represent physical material. Hence, information flow is not tied to material flow. Task flow --the initiation and completion of tasks-- is central. It would not be unusual for the modeller to include a number of dummy task nodes in a task chart; these dummy nodes would describe tasks whose counterparts are not even physical processing steps but rather scheduled delays or information passing activities. In addition, these nodes may or may not have material quantities associated with them. For example, one could translate a single order to produce at a certain production rate for a month (\( X \) units per day) into 20 different daily production orders, one for each working day of the month as follows: consider 20 dummy nodes, each with a built-in processing time that will delay its completion until the appropriate day of the month, and
consider another dummy node with a push edge emanating from it to each of the twenty delay nodes. Figure 9 illustrates. If this latter node is triggered with quantity X, each of the delay nodes will also be triggered with quantity X and can in turn be used to trigger a task sequence that will simulate the progress of a production order issued for the matching day. In this case, information in the form of production order quantities is being passed from task to task, but there is no physical material flow being simulated in these first steps of the task chart.

2.10 Inspection
Probabilistic edges were introduced as a way of modeling rework due to process failure. The material flow interpretation of probabilistic edges is that the material from the triggering task flows out along one and only one probabilistic edge. This is appropriate in the case of process failure but does not capture the situation in which material is inspected, defective units are pulled and the good units go on to further processing. For this purpose, the simulator recognizes inspect nodes: when a task associated with an inspect node completes, the quantity in the batch is split among all of the probabilistic edges emanating from the node and sent down the respective edges to trigger subsequent tasks. The quantity split is generated according to a multinomial distribution based on the probability weight attribute of the edges.

2.11 Part Numbers
At certain points in the production process, material, in certain forms, is recognized explicitly: reporting systems will keep track of inventories of certain part or product numbers. To mimic this, the simulator recognizes draw nodes and store nodes. Associated with each draw and store node is a part number. When the task associated with a store node (draw node) completes (starts), the simulator will increment (decrement) counters that track the inventory of that part number by the quantity in the batch associated with the task. Time plots and statistics of the dynamic behavior of these inventories are fundamental in the analysis of simulation output. As illustrated in Figures 10a and 10b, the modeller can
FIGURE 9. Using Task Nodes and Edges for Information Flow (Edge labels are transfer coefficients)
FIGURE 10a. Task Chart to Capture Intermediate Inventory
FIGURE 10b. PSD Capturing Intermediate Inventory
insert a store node-draw node combination anywhere in a task chart, assign a dummy part number to the
combination and thereby collect data on the flow of material over that point in the process. Usually, in
actual manufacturing systems, material in the process of being converted from one part number to
another is not tracked explicitly; its whereabouts can be deduced by studying work authorizations and by
tracing the material through the completed stages of the job. Hence, the tracking of material in the
COSMOS simulator (explicit for part numbers, implicit within task sequences) is quite similar to actual
systems.

2.12 Unitized vs. Batch Flow Simulation

Accounting for material flow is a chief feature of the simulator, and the above notions of units of
measure, transfer coefficients, transfer batches, batch sizes, processing time switches, inspections, and
part numbers are designed to allow the modeller to describe material flow relationships as closely as
possible. However, there are situations which the simulator is not designed to handle. For example, in
some manufacturing systems, individual units of material have serial numbers. Here, the logistics
information system tracks the progress of these serial numbers through the manufacturing process.
While such a level of detail is a powerful aid to manufacturing control, a unitized flow simulator (one
that tracks serial numbers) would seem to be overkill in a planning and simulation stage of analysis.
That is, most questions that are posed in planning and simulation can be answered in the aggregate
("expected number of units completed" rather than "which units completed and with what likelihood") so
that a batch flow simulator would be adequate.

2.13 Resources

Tasks consume the time of resources: machines, labor, and material storage and handling systems. Tasks,
with the exception of tooling, do not consume resources; they only consume the productive time in
which the resource is available. Work queues accumulate whenever tasks are in competition for a
common resource. The concept of a resource can be as general as an entire manufacturing plant or as
specific as a particular tool in a tool carousel. Simulating the progression of tasks requires only a mapping of tasks to resources and some measure of the resource’s capacity. Here, we distinguish between two types of capacity: task capacity and storage capacity. Depending on the level of aggregation, a resource can be single-tasking or multi-tasking (i.e. an assembly worker can work on only one task at a time). But, a department of fifty assembly workers could potentially process potentially fifty different tasks simultaneously. Task capacity is simply the number of different tasks the resource can process at the same time. Storage capacity is the amount of material, measured in whatever units the modeller chooses, the resource can hold. If a resource is full of material, it cannot accept any more tasks (draw tasks excepted), regardless of its task capacity. Note that there is no mention of production rates associated with a resource. In many manufacturing environments, equipment is characterized by a relatively fixed production rate (measured in units of product per unit time). However, even in these environments, some products take longer than others on the same equipment. Thus, generally we associate the specification of production times with the task using the resource rather than with the resource itself. The attributes of a task node include a list of the resources required and a specification of the rates at which the capacity of each resource is consumed (i.e. the task and storage size of the task relative to the task and storage capacity of the resource). Task size can be set based on the number of units in a batch or independently of batch size. Storage size is strictly related to the number of units in the batch. The material is assumed to reside in only one of the resources in the list of requirements. This is designated as the primary resource for the task. Every task node must include specification of a primary resource, but if the task is not meant to represent a physical processing step, the task size and storage size can be set to zero to minimize the interaction of the task with the resource. Task capacity is consumed by tasks in progress according to the task size of the batch. Storage capacity is consumed by tasks in progress according to the storage size of the batch, by transfer batches awaiting the beginning of subsequent tasks, and by inventory (material that has been stored according to store tasks but has not yet been drawn by draw tasks).
2.14 **Resource Types and Zones**

Resources are classified by type and zone. Resources within the same resource type have identical processing characteristics: the same task capacity and storage capacity, the same time-to-failure distribution, and so on. Zones are groups of resources that correspond to physical groupings on the factory floor. For example, a department may consist of several machines, attendant workers, a setup crew, and material handling equipment. The narrowest zone definition is a single resource (e.g. a specific piece of equipment). The broadest definition is the entire set of resources. In specifying resource requirements, task nodes must identify a resource type and a resource zone (for example "milling machine in machine cell A"). The resource request can be satisfied by any resource matching the specification: for example, by any milling machine in cell A.

2.15 **Processing Continuity**

When one task follows another and both tasks have specified the same primary resource requirement, the modeller's intent would likely be that the same resource should be used to satisfy both resource requests. The simulator attempts this matching using the transfer batch record. When a task completes and a transfer batch record is created, it is associated with the resource that was assigned to satisfy the primary resource request of the now completed task. In this way, the simulator remembers what resources were used to satisfy requests. When it comes time to satisfy the primary resource request of the triggered task, the transfer batch is checked first to see if the resource recorded there will suffice. If it will, then it will be assigned to the triggered task, and processing will appear to be continuous on the same resource. Otherwise, any other resource matching the specification will be used.

2.16 **Changeover Times and Setup Tasks**

Equipment setups can be modelled as tasks whose time is specified on a per batch basis, independent of the number of units in the batch. This is sufficient for many manufacturing environments, but there are situations when changeover time depends on what task or product was last run on the equipment. Changing from yellow paint to black paint in a paint booth takes less time to clean out the hoses than
changing from black paint to yellow. To be general, we think of the resource as being in a state, such as BLACK, YELLOW, GREEN, etc., and that certain tasks, indicated by setup nodes, can change the resource to a new state. The setup nodes indicate the new state and the minimum changeover time. The name of the state is created by the modeller. Associated with the resource type is a table of additional changeover times, one for each combination of old state and new state. If the resource assigned to the setup task is already in the desired state, then the setup time is taken to be zero. Otherwise, the setup time is the sum of the minimum changeover time from the setup node and the additional changeover time of the resource type. Both changeover times, minimum and additional, are specified as random variables.

2.17 The Resource Chart

There is a considerable amount of information associated with resources, distinct from the resource requirements that are listed as attributes of the task nodes. This information is organized in a two level structure. The top level is simply a text file of resource type records. This contains all the basic data that describes the processing characteristics of each resource type: task capacity, storage capacity, time to failure distribution, time to repair distribution and changeover time table. The next level is a graphical representation of the factory resources, the resource chart: each node represents a particular resource. Zones are sets of nodes and can be indicated graphically in a variety of manners (highlighting, color, "lassos," etc.). The attributes of the resource node indicate the resource type of the resource, the current state of the resource, the current tasking utilization, the current storage utilization and the current list of contents (part numbers).

2.18 Dispatching

When tasks are initiated, they give rise to resource requests and, in the case of draw tasks, to material requests. A task cannot begin until all its resource and material requests have been satisfied. Tasks can be in competition for the same resources and materials so there is a need for an allocation
mechanism. Dispatching refers to the simulated activity of allocating scarce resources and materials among competing requests. There are countless rules and algorithms that could be implemented to perform dispatching. Classical rules include First-Come-First-Serve, Shortest-Processing-Time, and Earliest-Runout. Refer to Conway, Maxwell and Miller [1967] for seminal work in this area. One purpose of the COSMOS simulator is to allow for experimentation with different dispatching rules. The researcher or ambitious modeller can program alternative dispatching routines and link them into the simulator. Research into intelligent dispatching routines is an on-going activity at Cornell. These routines will be included in a software library subsidiary to the simulator as they become available.

2.19 Scheduling and Task Chart Simulation

As a simulation language, the task chart is a powerful tool, capable of representing a myriad of manufacturing situations. Its strength is that it can represent complex sequences of tasks, with their associated material flow and resource requirement implications in a concise and graphical form. Its chief limitation is that individual tasks are not activities which can be scheduled: they are triggered according to a pre-specified sequence implied by the task chart. To link simulation with scheduling, we introduce the concept of an operation, defined to be a task chart with a single root node. An operation can be scheduled by means of a lot release which specifies a release time and release quantity. In a simulation run, the root node of the task chart is triggered with the release quantity at the release time. The tasks that are spawned from the root node and its successors in the task chart are collectively referred to as the job or lot associated with the lot release. When all of the spawned tasks have completed, the job is said to be completed. A lot release is analogous to releasing a work authorization to the shop floor. The decision to release the work is the prerogative of the scheduler; once released, it is the responsibility of the dispatcher to monitor the progress of the job and allocate the resources necessary for the job to complete. Thus, tasks are the fundamental elements for simulation; jobs or lots are the fundamental elements for scheduling. A series of lot releases for an operation constitutes a production
schedule for that operation. Instead of specifying one large task chart to describe all the activities that take place in the factory, the modeller groups the activities into operations and provides a production schedule or a scheduling algorithm to the simulator. This grouping is a fundamental decision that the modeller faces. The larger the separate task charts, the less control over the timing of individual tasks the modeller can exercise and the greater the responsibility of the dispatcher. The smaller the task charts, the more operations the scheduler must manage.

2.20 **Product Structure Diagram**

Operations are related to other operations by means of part numbers. That is, one task chart may contain a store node with a certain part number and another task chart may contain a draw node with the same part number. Material flow from the one operation to the other is implied. The **product structure diagram** is a graph that summarizes these relationships. The nodes of the product structure diagram consist of **operation nodes** and **inventory nodes**. There is a one to one correspondence between operation task charts and operation nodes and between part numbers and inventory nodes. The edges of the graph are referred to as **bill of materials edges** if they emanate from an inventory node and terminate in an operation node. **Yield edges** emanate from an operation node and terminate in an inventory node. There is a one to one correspondence between bill of materials edges in the product structure diagram and draw nodes in the operation task chart and between yield edges and store nodes. Figure 11 illustrates. The key attribute of a bill of materials edge is the bill of materials quantity: the number of units of the part number required per unit of the operation. The key attribute of a yield edge is the planned yield: the expected number of units of the part number created per unit of the operation. (The units of measure of the operation are the units of material measure for the root node of the operation task chart.) Both attributes can be estimated by analyzing the underlying operation task chart. Just as the task chart is the framework for the simulation database, the product structure diagram can be viewed as the framework of a scheduling database. In general, production scheduling algorithms will ignore the detailed description of manufacturing steps captured by the operation task
FIGURE 11. Correspondence of PSD Edges and Task Chart Nodes

PART1

BOM EDGES

DRAW_PART1

STORE_PART2

OPN1

DRAW_PART1

STORE_PART2

PART2

YIELD EDGES
charts, and instead, operate on the summarized relationships of the product structure diagram. Similarly, the physical inventory of a particular part number can be scattered over a number of different resources (stores areas, staging areas, work in process buffers, etc.). The simulator tracks this inventory by maintaining a contents list against each resource. It also maintains (as an attribute of the inventory nodes) the total inventory of each part number across all resources. Scheduling algorithms will ignore the detailed contents lists and generate production orders based on present and future total inventory levels.

2.21 Static vs. Dynamic Scheduling

One possible application of this modelling environment is to evaluate a production schedule. The user could set the inventory levels in the model to match the current inventory levels in the factory and let the simulation proceed from that state with operations being triggered according to the production schedule. Thus, the user could forecast resource utilization and inventory levels over the duration of the schedule. Run controls in the simulator permit different global assumptions such as "assume all processing times are deterministic (non-probabilistic)," "ignore storage capacity restrictions," "ignore equipment failures," and others. The user can make a variety of such runs to assess the impact of the schedule and make changes to the schedule to improve the results. As the actual results of the production process are reported from the factory, they could be compared against planned or simulated results and corrective action could be taken. In this mode, the production schedule is static as far as the simulator is concerned. Another application of the modelling environment would be to investigate scheduling algorithms. Different algorithms can have dramatically different effects on system performance. Consequently, simulation studies commissioned to assess the performance characteristics of proposed manufacturing system designs should consider not only alternative system designs but also alternative scheduling regimens. Evaluation of scheduling algorithms in the context of frequently observed system designs is the prime use of COSMOS as a research tool. For such studies, the scheduling algorithm is imbedded in the simulator. The algorithm's dynamic behavior in response to
changing conditions can then be simulated. Comparative studies of different algorithms for different manufacturing systems is a rich area for research. Hybrid approaches in which certain operations are controlled by scheduling algorithms and other operations are controlled by pre-specified production schedules are also possible within this framework.

2.22 Different Algorithms Require Different Data

All information contained in the task chart is crucial to the COSMOS simulator: task times, material quantities, resource requirements, trigger relationships, etc. Identifying the required additional information for scheduling at the level of the product structure diagram is more problematic because of the variety of approaches to production scheduling that are possible. We have identified an initial set of attributes to be maintained at the nodes of the product structure diagram that should be sufficient to support several alternative scheduling approaches. Some of these attributes must be maintained by the simulator (such as inventory levels at the inventory nodes). Other attributes are specified by the user (such as policy parameters). For example, a "pull" or replenishment type scheduling algorithm requires the specification of a reorder point and a reorder quantity for each part number. These are policy parameter attributes of the inventory node. They are to be maintained by the user, perhaps with the aid of external optimization algorithms for determining good parameter values. In turn, such optimization algorithms would require access to additional parameters such as product cost (an attribute of the inventory node that is also the responsibility of the user). On the other hand, a "push" or produce-to-plan type algorithm, such as Material Requirement Planning (MRP), requires an estimate of the processing time or production lead time of each operation. To be general, the attributes of the operation node (distinct from the task chart associated with the operation node) include both a fixed and a variable component estimate of processing time: estimated processing time is the sum of the fixed time and the number of units in the lot release multiplied by the variable time. These estimates can be maintained by routines that analyze the underlying task chart. As another example, capacitated production planning algorithms such as Jackson, Maxwell and Muckstadt [1986] need some summary of
the resource requirements of the underlying task chart. This summary is an attribute of the operation node. Undoubtedly, the list of attributes for nodes in the product structure diagram will grow as more and more scheduling algorithms are added to the library. For example, against each inventory node it would be appropriate to attach a forecast demand and a forecast supply schedule. The documentation of each scheduling algorithm must indicate which attributes are relevant and which can be ignored.

2.23 **Tasks, Operations and Processes**

MRP systems are frequently organized around the concepts of part numbers, for which production orders are placed, bills of materials, which describe the input part numbers and quantities required for orders, and routing sheets, which describe the sequence of steps required to transform the inputs into the output part numbers. One way to transform this information into the COSMOS databases is to think of the routing sheet as a model for the task chart, with the tasks being the individual steps listed in the routing sheet. This approach implies that the scheduler will have little control over the progress of a production order; it is in the hands of the dispatcher. To exercise scheduling control, the modeller can break up the routing sequence into individual operations, a separate task chart for each step in the sequence. The material between these operations must be assigned a part number for the simulator to manage. These part numbers will not match up with any part number in the MRP system so we distinguish between two types of inventory nodes in the product structure diagram: **stock nodes**, MRP-type part numbers for which some inventory is maintained, and **transit nodes**, intermediate part numbers for which inventory should only exist for very short periods of time. The bill of material structure can be deduced quite readily from the product structure diagram. From any stock node, it is a simple matter to trace backward along yield and bill of material edges, past the transit nodes, in order to identify all the stock nodes that are required as input. Furthermore, the stock nodes can be used to partition the product structure diagram into subgraphs that consist of operations and transit nodes only; these subgraphs are called **processes**. A process relates a set of input stock nodes to a set of output
stock nodes. Figure 12 illustrates. Scheduling algorithms can be constructed around the notion of processes and, within processes, operations. The hierarchy is thus:

```
scheduled processes
  ↓
scheduled operations
  ↓
triggered tasks
```

2.24 Importance of the User Interface

This concludes our overview of the COSMOS database. The main intent in designing the database structure was to ensure that the representation capabilities would be adequate for the full range of uses to which COSMOS would be applied. The result is a very rich modelling system that can form the basis for a wide variety of applications ranging from industrial feasibility studies to algorithmic research experiments. Given the complexity of the database, the importance of the user interface becomes paramount. If the application is simple, the modeller should be able to create a simple model and analyze it quickly. COSMOS provides this capability.
FIGURE 12. Use of Transit Nodes to Define Processes of More Than One Operation

Process for P2

Process for P4

Process for P5 and P6
References


"Good Medicine or Bad." Business Week, Nov. 10, 1986, p. 86.


"Invasion USA." *Business Week*, Nov. 10, 1986, p. 84.


**SIMFACTORY,** CACI Products Company, 3344 North Torrey Pines Court, La Jolla, California, 92037.


Chapter 3
COSMOS USER MANUAL DESCRIPTION

3.1 Overview

The objective of this manual is to describe how to create, view and edit a COSMOS factory simulation model, how to simulate it, and how to interpret the results.

3.2 Related Documents

In addition to this manual, the following manuals provide an in-depth view of COSMOS.

3.2.1 COSMOS Data Structure: Written in October 1986, this provides an introduction to the basic concepts of case, graph, node and edge used in COSMOS data structure.

3.2.2 COSMOS Flatfile Description: Written in May 1987, this manual provides an introduction to the data structure of Resource Chart, Product Structure Diagram, and Task Chart.

3.2.3 COSMOS Screen Manager: Written in January 1987, this manual provides an introduction to the functions to the screen manager program and the sub procedures.

3.3 Notes on this Manual

In the COSMOS User’s Manual, there are three different fonts:

This is Times Roman standard

This is Times Roman bold

This is Helvetica standard

Key words, concepts, terms and menu selections are printed in Times Roman bold. In addition, there are a number of terms which are printed in Times Roman bold to distinguish them from the rest of the text. For example:

When discussing from nodes, the word from is printed in Times Roman bold to alert the reader that it is a specific type of node instead of a preposition.

All screen output from COSMOS (i.e. warning messages, prompts) are printed in Helvetica standard.

The "normal" font for the COSMOS User’s Manual is Times Roman standard.
Chapter 4
GENERAL GRAPHICAL EDITING FUNCTIONS

This chapter will introduce to you the following features of the COSMOS graphical editing system:

1. GENERAL FUNCTIONS acquaints the user with the screen format of the COSMOS menu.
2. MAIN MENU explains the selections the user can make to enter the other sub-menus.
3. PRODUCT STRUCTURE DIAGRAM MENU teaches the user how to create, view or edit the Product Structure Diagram of a case.
4. TASK CHART MENU shows the user how to create, view or edit the Task Chart of an operation node which belongs to the Product Structure Diagram of a case.
5. RESOURCE CHART MENU shows the user how to create, view or edit the Resource Chart of a case.

All COSMOS graphical editing menus have a title area, a window area, a message area and a user area. Each view has a small frame around it called the View Menu. The View Menu has buttons for translating, scaling, deleting, and reordering that view.

4.1 Hardware Requirements

COSMOS requires an IBM 5080-2 graphics terminal, an IBM 5083 graphics tablet, and an IBM 3270 display terminal. In addition, COSMOS needs an IBM 4341, or equivalent VM370 machine. Finally, the user will need access to enough disk storage to accommodate the COSMOS test cases and simulation output. Generally, 20 cylinders of disk storage should be sufficient.

4.2 Software Requirements

COSMOS runs under IBM CMS with a virtual machine storage size of 7 megabytes. In addition, there are other software packages which COSMOS needs to have available to it (for example, GAM/SP and GDDM). The user's account should already be properly configured to run COSMOS.

Before running COSMOS, the user must attach the 5080 graphics terminal to the account. The procedure to attach the terminal varies from installation to installation. After the graphics terminal has been attached, type:

COSMOS

The program requires a few moments to initialize itself. When COSMOS has been initialized, the main menu screen will appear on the graphics terminal. From the main menu, select one of the items which appears on the right side of the screen.

4-1
4.3 COSMOS Overview

There are two main parts to COSMOS: graphical editing and data editing. Graphical editing, on the 5080 graphics terminal, defines the logical structure of a COSMOS model (e.g. how one operation is associated with another). Data editing is performed on the 3270 data terminal and specifies numeric values to different parts of a model (e.g. the capacity of a machine). When the user adds an icon to a model, that icon will have certain default values associated with it. The data editor is used to change these values to match the requirements of your particular model.

4.4 Using the 5080 Graphics 'Mouse'

The 5080 graphics mouse is a graphical selection device wired directly to the 5080 graphics tablet. As the mouse moves across the tablet, a graphics cursor will appear on the graphics terminal. To select an item displayed on the graphics screen, position the graphics mouse cross hairs over the screen item and press any one of the mouse’s four function keys to highlight this entry. The lettering will turn from white to red to confirm that the option is now in effect.

4.5 Translating A View

The view box can be moved to a different part of the graphics screen. First, point to the top bar of the view menu, then point the pen to any other location in the user area. The view box will then be moved to the new location.

4.6 Scaling A View

The size of the view box can also be changed. First, point to the bottom bar of the view. Then point the selector to any other location in the user area. The view will be redrawn with the new size.

4.7 Reordering Views:

If the user selects the right bar, the view list will be reordered. The selected view will be moved to the end of the view list.

4.8 Deleting A View

Selecting the left bar of a view will delete the view.

Note: If the selected view is a Resource Chart or a Task Chart, then only that view will be deleted. If the selected view is a Product Structure Diagram, all the graphs associated with the case will be deleted.
4.9 The Active View

Only one view at a time is active. The active view is denoted by a red frame around the display window. A view is activated by pointing to it (using the graphics mouse) and pressing one of the selector function keys. All other frame colors (non-active) are green. The graphics options menu to the right of the screen will correspond to options relating to the active view. When a view is activated, the view border will change from green to red, and it will be brought to the front of the screen display if it had previously been obstructed by another view.
Chapter 5

MAIN MENU

On the Main Menu, the user can make any of the following selections when the main view is active.

- inventory
- mat cy lo
- mat cy up
- net inven
- back log
- shortfall
- resc util
- lot rel
- editor
- help
- exit
- simulator

5.1 Inventory

A. After inventory is selected, the inventory history selection menu will appear on the data entry terminal screen. At this time, the user can display one of the existing inventory history files. PF3 will cancel the selection menu.

B. To display an existing inventory history file, move the cursor to a file name using either the TAB or cursor keys and press the ENTER key. COSMOS will then display the message:

   select part numbers to be plotted

C. After the part number selection menu appears on the data entry terminal screen, the user may display an inventory plot for selected part numbers. PF3 will cancel this menu.

D. To display the inventory plot, position the cursor over a part number using either the TAB or cursor keys and press the ENTER key to select it. Selected part numbers will be highlighted. The user may select more than one part number. COSMOS will then display the message:

   please wait for plot....

E. The plot will appear in a window at the upper left side of the user area, and the inventory menu will become the active menu.

F. Here, the user may choose to examine the data for a specific point in the inventory plot. To activate this feature, position the mouse selector cross-hairs over the center of the point, and press one of the mouse function keys. COSMOS will display the data for that point.

G. From the inventory menu screen, the inventory plot view is the active view. The user can make any of the following selections from the menu. Further information on functions not described below (i.e. inventory, mat cy lo, mat cy up and net inven) can be found in subsequent chapters.

   - pan left: The window is moved 20% to the left. This feature displays more of the right part of the inventory plot in the active window.
   - pan right: The window is moved 20% to the right. This feature displays more of the left part of the inventory plot in the active window.
   - zoom in: The range of the x-axis in the inventory plot is decreased by 20%

5-1
zoom out  The range of the x-axis in the inventory plot is increased by 20%.
reset  The inventory plot is reset back to the original scale.
help  The help menu is not implemented.
exit  Exit to main menu.

5.2 Mat Cy Lo

After mat cy lo is selected, an inventory history selection menu will appear on the data entry terminal screen. The user can either display one of the existing inventory history file, or hit PF3 to exit the selection menu.

A. To display an existing inventory history file, move the cursor over a file name using either the TAB or cursor keys and press the ENTER key.

B. The test case selection menu will appear on the data entry terminal screen. COSMOS now expects the user to select the associated case file. COSMOS will display the message:

select part number to be plotted

A part number selection menu will appear on the data entry terminal screen.

C. The user can either display the mat cy lo plot for a selected part number or hit PF3 to exit the menu.

D. To display the mat cy lo plot, move the cursor over a part number using either the TAB or cursor keys and press the ENTER key. COSMOS will display the message:

please wait for plot....

E. The mat cy lo plot will now appear at the upper left side of the user area, and the mat cy lo menu will become the active menu.

F. The user may examine the data of a specific point in the mat cy lo plot. Position the mouse selector cross hairs over the center of the point, and press one of the mouse function keys. COSMOS will then display the data for that point.

G. On the mat cy lo menu screen, the mat cy lo plot view is the active view. The user can select any of the following options. For further information on commands not described below (i.e. inventory, mat cy lo, mat cy up, net inven, back log and shortfall), please refer to subsequent chapters.

pan left  The window is moved 20% to the left. This feature displays more of the right part of the mat cy lo plot in the active window.

pan right  The window is moved 20% to the right. This feature displays more of the left part of the mat cy lo plot in the active window.

zoom in  The range of the x-axis in the mat cy lo plot is decreased by 20%.
5.3 Mat Cy Up

A. After mat cy up is selected, an inventory history selection menu will appear on the data entry terminal screen.

B. The user can either display one of the existing inventory history files, or hit PF3 to exit the selection menu.

C. To display an existing inventory history file, move the cursor to a file name using either the TAB or cursor keys and press the ENTER key.

D. The test case selection menu will appear on the data entry terminal screen. At this point, the user must now select the associated case file. COSMOS will display the message:

    select part number to be plotted

A part number selection menu will appear on the data entry terminal screen.

E. The user may now display the mat cy up plot for a selected part number. PF3 cancels this menu. To display the mat cy up plot, move the cursor over to a part number using either the TAB or cursor keys and press the ENTER key. COSMOS will display the message:

    please wait for plot.....

F. The mat cy up plot will appear at the upper left side of the user area, and the mat cy up menu will become the active menu.

G. To see the data of a specific point in the mat cy up plot, position the mouse selector cross hairs over the center of the point, and press one of the mouse function keys. COSMOS will display the data for that point.

H. On the mat cy up menu screen, the mat cy up plot view is the active view. The user can select any of the following options. For further information on commands not described below (i.e. inventory, mat cy lo, mat cy up, net inven, back log and shortfall), please refer to subsequent chapters.

    pan left    The window is moved 20% to the left. This feature displays more of the right part of the mat cy up plot in the active window.

    pan right   The window is moved 20% to the right. This feature displays more of the left part of the mat cy up plot in the active window.

    zoom in     The range of the x-axis in the mat cy up plot is decreased by 20%.
zoom out The range of the x-axis in the mat cy up plot is increased by 20%.
reset The mat cy up plot is reset back to the original scale.
help The help menu is not implemented.
exit Exit to main menu.

5.4 Net Inventory

A. After net inven is selected, an inventory history selection menu will appear on the data entry terminal screen.

B. The user can now display one of the existing inventory history files, or hit PF3 to exit the selection menu.

C. To display an existing inventory history file, move the cursor over a file name using either the TAB or cursor keys, and press the ENTER key. COSMOS will display the message:

select part numbers to be plotted

A part number selection menu will appear on the data entry terminal screen.

D. The user can either display the net inventory plot for the selected part numbers, or hit PF3 to exit the menu. To display the net inventory plot, move the cursor over a part number using either the TAB or cursor keys, and press the ENTER key. By repeating the selection process, the user may display the plot for more than one part number. COSMOS will display the message:

please wait for plot.....

E. The net inventory plot will appear at the upper left side of the user area, and the net inventory menu will become the active menu.

F. To see the data of a specific point in the net inventory plot, position the mouse selector cross hairs over the center of the point, and press one of the mouse function keys. COSMOS will display the data for that point.

G. On the net inventory menu screen, the net inventory plot view is the active view. The user can select any of the following options. For further information on commands not described below (i.e. inventory, mat cy lo, mat cy up, net inven, back log and shortfall), please refer to subsequent chapters.

pan left The window is moved 20% to the left. This feature displays more of the right part of the net inventory plot in the active window.

pan right The window is moved 20% to the right. This feature displays more of the left part of the net inventory plot in the active window.

zoom in The range of the x-axis in the net inventory plot is decreased by 20%.
zoom out  The range of the x-axis in the net inventory plot is increased by 20%.
reset  The net inventory plot is reset back to the original scale.
help  The help menu is not implemented.
exit  Exit to main menu.

5.5 Back Log

A. After back log is selected, an inventory history selection menu will appear on the data entry terminal screen.

B. The user can either display one of the existing inventory history files, or hit PF3 to exit the selection menu.

C. To display an existing inventory history file, move the cursor over a file name using either the TAB or cursor keys and press the ENTER key. COSMOS will display the message:

   enter the name of a shipping node or blank to exit

   The user can type in the name of the shipping node to be displayed, or enter a blank string to exit.

D. The back log plot will appear at the upper left side of the user area, and the back log menu will become the active menu.

E. To see the data of a specific point in the back log plot, position the mouse selector cross hairs over the center of the point, and press one of the mouse function keys. COSMOS will display the data for that point:

F. On the back log menu screen, the back log plot view is the active view. The user can select any of the following options. For further information on commands not described below (i.e. inventory, mat cy lo, mat cy up, net inven, back log and shortfall), please refer to subsequent chapters.

   pan left  The window is moved 20% to the left. This feature displays more of the right part of the back log plot in the active window.

   pan right  The window is moved 20% to the right. This feature displays more of the left part of the back log plot in the active window.

   zoom in  The range of the x-axis in the back log plot is decreased by 20%.

   zoom out  The range of the x-axis in the back log plot is increased by 20%

   reset  The back log plot is reset back to the original scale.

   help  The help menu is not implemented.

   exit  Exit to main menu.
5.6 **Shortfall**

A. After shortfall is selected, an inventory history selection menu will appear on the data entry terminal screen.

B. At this time, the user can either display one of the existing inventory history files, or hit PF3 to exit the selection menu.

C. To display an existing inventory history file, move the cursor over a file name using either the TAB or cursor keys and press the ENTER key.

D. The material activity record selection menu will appear on the data entry terminal. COSMOS now expects the user to select the associated file, or hit PF3 to exit the selection menu.

E. After a file is selected, a shipping node selection menu will appear on the data entry terminal screen. The user can display the shortfall plot for a selected shipping node, or hit PF3 to exit the menu.

F. To display the shortfall plot, move the cursor to a shipping node using either the TAB or cursor keys and press the ENTER key. COSMOS will display the message:

```
You may offset planned ship dates in the plot relative to the actual ship dates
Enter offset in time:
```

G. After the user inputs a number, the shortfall plot will appear at the upper left side of the user area, and the shortfall menu will become the active menu.

H. In order to see the data of a specific point in the shortfall plot, position the mouse selector cross hairs over the center of the point, and press one of the mouse function keys. COSMOS will display the data for that point.

I. On the shortfall menu screen, the shortfall plot view is the active view. The user can select any of the following options. For further information on commands not described below (i.e. inventory, mat cy lo, mat cy up, net inven, back log and shortfall), please refer to subsequent chapters.

- **pan left** The window is moved 20% to the left. This feature displays more of the right part of the shortfall plot in the active window.

- **pan right** The window is moved 20% to the right. This feature displays more of the left part of the shortfall plot in the active window.

- **zoom in** The range of the x-axis in the shortfall plot is decreased by 20%.

- **zoom out** The range of the x-axis in the shortfall plot is increased by 20%.

- **reset** The shortfall plot is reset back to the original scale.

- **help** The help menu is not implemented.

- **exit** Exit to main menu.
5.7 Resc Util

A. When recus util is selected, a resource utilization selection menu will appear on the data entry terminal screen.

B. At this time, you can either display one of the existing resource utilization files, or hit PF3 to exit the selection menu.

C. To display an existing resource utilization file, move the cursor to a file name using either the TAB or cursor keys, and press the ENTER key.

D. At this point, the resource utilization plot will appear at the upper left side of the user area, and the resc util menu will become the active menu.

E. On the shortfall menu screen, the shortfall plot view is the active view. The user can select any of the following options. For further information on commands not described below (i.e., inventory, matcylo, matcyup, netinven, back log and shortfall), please refer to subsequent chapters.

- **new plot**
  This function performs the same effect as the resc util in the main menu screen.

- **pan left**
  The window is moved 20% to the left. This feature displays more of the right part of the resc util plot in the active window.

- **pan right**
  The window is moved 20% to the right. This feature displays more of the left part of the resc util plot in the active window.

- **zoom in**
  The range of the x-axis in the resc util plot is decreased by 20%.

- **zoom out**
  The range of the x-axis in the resc util plot is increased by 20%.

- **page up**
  The window is moved one page upward, so that the previous page of the resource utilization plot can be shown in the active window.

- **page down**
  The window is moved one page downwards, so that the next page of the resource utilization plot can be shown in the active window.

- **reset**
  The resource utilization plot is reset back to the original scale.

- **report**
  The report menu has not yet been implemented.

- **help**
  The help menu has not yet been implemented.

- **exit**
  Exit to main menu.

5.8 Lot Rel

See the chapter on the Lot Release Editor.
5.9 Editing a Test Case

To edit a Test Case, select the editor option from the main menu. A case selection menu will appear on the data entry terminal screen. The user can now display one of the existing cases or display a new case.

To display an existing case:

Move the cursor over a case name using either the TAB or cursor keys, and then press the ENTER key. The Product Structure Diagram (PSD) of that case will be displayed as an active view, and the PSD menu will become the active menu.

To create a new case:

Press the PF3 to edit the case selection menu. COSMOS will display the message:

`please enter a new case name (8 chars maximum) = = >`

Type in the name of the new case. Invalid case names include blank strings and case names which already exist. An invalid case name will cause COSMOS to prompt the user for a new case name. After the user has entered a valid case name, a blank Product Structure Diagram (PSD) view will appear at the upper left side of the screen, and the PSD menu will become the active menu.

5.10 Simulator

Refer to Chapter 10.

5.11 Help

Invokes the help menu.

5.12 Exit

Exits the menu.
Chapter 6
PRODUCT STRUCTURE DIAGRAM MENU

On the Product Structure Diagram menu, the Product Structure Diagram view is the active view. From here, the user can select any of the following choices:

<table>
<thead>
<tr>
<th>Action</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>pan left</td>
<td>edit node</td>
</tr>
<tr>
<td>pan right</td>
<td>move node</td>
</tr>
<tr>
<td>pan up</td>
<td>del node</td>
</tr>
<tr>
<td>pan down</td>
<td>goto node</td>
</tr>
<tr>
<td>zoom in</td>
<td>add edge</td>
</tr>
<tr>
<td>zoom out</td>
<td>view edge</td>
</tr>
<tr>
<td>reset</td>
<td>edit edge</td>
</tr>
<tr>
<td>res chrt</td>
<td>del edge</td>
</tr>
<tr>
<td>task chrt</td>
<td>shipping</td>
</tr>
<tr>
<td>view node</td>
<td>receiving</td>
</tr>
<tr>
<td>transit</td>
<td>stock</td>
</tr>
<tr>
<td></td>
<td>operation</td>
</tr>
<tr>
<td></td>
<td>label</td>
</tr>
<tr>
<td></td>
<td>simulator</td>
</tr>
<tr>
<td></td>
<td>get case</td>
</tr>
<tr>
<td></td>
<td>save</td>
</tr>
<tr>
<td></td>
<td>rename</td>
</tr>
<tr>
<td></td>
<td>file</td>
</tr>
<tr>
<td></td>
<td>help</td>
</tr>
</tbody>
</table>

Descriptions of all the Product Structure Diagram functions follow:

6.1 **Pan Left**

The Product Structure Diagram is moved 20% to the right in order to display the left side of the Product Structure Diagram.

6.2 **Pan Right**

The Product Structure Diagram is moved 20% to the left in order to display the right side of the Product Structure Diagram.

6.3 **Pan Up**

The Product Structure Diagram is moved 20% downward in order to display the top of the Product Structure Diagram.

6.4 **Pan Down**

The Product Structure Diagram is moved 20% upward in order to display the bottom of the Product Structure Diagram.

6.5 **Zoom In**

The range of the x-axis in the Product Structure Diagram is decreased by 20%.
6.6 **Zoom Out**

The range of the x-axis in the Product Structure Diagram is increased by 20%.

6.7 **Reset**

The Product Structure Diagram is reset back to the original scale.

6.8 **Res Chrt**

When selected, **res chrt** displays the Resource Chart of the test case as an active view. The Resource Chart menu, described later, becomes the active menu.

6.9 **Task Chrt**

A. When **task chrt** is selected, COSMOS will respond with the message:

    task chart: select an operation node

    Here, the user must choose the operation node type (shipping or receiving). Before making a selection, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

B. After the user has selected an operation node type, the task chart will appear as an active view. In addition, if the Task Chart Menu was not displayed previously, it will become the active menu.

6.10 **View Node**

A. When **view node** is selected, COSMOS will prompt the user to select a node. Before selecting a node, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

B. After the user has selected a node, the Product Structure Diagram Node Record Menu will appear on the data entry terminal screen. COSMOS will respond with the message:

    view node on the data entry terminal screen

    See the section for information on editing operation nodes.

6.11 **Edit a Node**

A. After **edit node** has been selected, COSMOS will display the message:

    edit node: select a node to edit

B. COSMOS now expects the user to select a node. Before selecting a node, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.
C. After the user has selected a node, the record of that node will be shown on the data entry terminal screen. COSMOS will display the message:

    edit node on the data entry terminal screen

For further information, see the manual section describing the editing of operation nodes.

6.12 Move Node

A. When move node is selected, COSMOS will display the message:

    move node: pick node to move

B. Before selecting a node to move, the user can pan or zoom the Product Structure Diagram, and move or scale any view box. After the user selects a node, COSMOS will display the message:

    move node: point to a grid location

C. The user must now point to the desired destination grid location. The node will then be moved to this new location. In addition, all the edges pointing from or into this node will follow the node to the new location.

    Note: If the new location is already occupied by another node, COSMOS will warn the user and request a new location.

6.13 Del Node

A. After del node is selected, COSMOS will display the message:

    delete node: pick node to delete.

B. Before selecting a node to delete, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. After the user has selected a node, COSMOS will delete that node. Likewise, all the edges pointing from and into the selected node will disappear.

6.14 Goto Node

A. After goto node is selected, COSMOS will display the message:

    select a psd node on the data entry terminal screen:

On the data terminal, COSMOS will display a menu showing all the Product Structure Diagram nodes. Use the TAB or cursor movement keys to move the cursor. The ENTER key will accept the current selection. After a node is selected, the Product Structure Diagram will be moved so that the selected node is in the center of the view box.
6.15 Add Edge

A. After add edge is selected, COSMOS will display the message:

   add edge: pick from node

B. Before selecting a from node, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. The user must now select a from node.

D. After selecting a from node (i.e. chipA), COSMOS will display the message:

   CHIPA selected, pick into node

E. Before selecting an into node, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

F. The user must now select an into node.

G. After selecting an into node (i.e. chipB), COSMOS will determine if there is an edge which joins the selected from node and the selected into node. If there is already such an edge, COSMOS will display the message:

   add edge: the edge is already existing

If the location is available, COSMOS will display the message:

   CHIPB selected

H. If the from node is an inventory node, and into node is an operation, shipping or receiving node, the type of the newly created edge will be Bill of Materials. In addition, if there is not yet a draw node referencing this part in the Task Chart of the selected (operation, shipping, receiving) node, one will automatically be created in the Task Chart.

I. If the into node is an inventory node, and the from node is an operation, shipping or receiving node, COSMOS will create a new yield edge. Also, if necessary, a store node will be created in the Task Chart of the selected (operation, shipping, or receiving) node. This store node will reference the selected into inventory node.

6.16 View Edge

A. After view edge is selected, COSMOS will display the message:

   view edge: pick edge to view only

B. Before selecting an edge to view, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select an edge.
D. The Product Structure Diagram Edge Editor appears on the data entry terminal screen. This editor lists the records of the selected edge. COSMOS will display the message:

        view edge on the data entry terminal screen

E. PF3 exits the editor.

6.17 Edit Edge

A. After edit edge is selected, COSMOS will display the message:

        edit edge: pick edge to change

B. Before selecting an edge to edit, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select an edge.

D. The Product Structure Diagram Edge Editor appears on the data entry terminal screen. The editor lists the records of the selected edge. COSMOS will display the message:

        edit edge on the data entry terminal screen

        Press the TAB or cursor movement keys to go to different line on the data terminal menu.

E. Press PF3 to exit the editor.

6.18 Del Edge

A. After del edge is selected, COSMOS will display the message:

        delete edge: pick edge to delete

B. Before selecting an edge to delete, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select an edge.

D. If the edge type is Bill of Materials (inventory --> operation, shipping or receiving) then the draw node in the task chart of the (operation, shipping, or receiving), which contains the part_no of the inventory node will be deleted. If the edge type is yield (operation, shipping or receiving --> inventory) then the store node in the task chart of the (operation, shipping, or receiving), which contains the part number of the inventory node will be deleted.
6.19 Shipping

A. After shipping is selected, COSMOS will display the message:
   create node: point to a grid location

B. Before pointing to a location, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select a grid location.

D. After a location has been selected, COSMOS will create a shipping node in that location.

   Note: If the location is already occupied by another node COSMOS will display the message:
   create node: this location is already occupied

6.20 Receiving

A. After the receiving menu option is chosen, COSMOS will display the message:
   create node: point to a grid location

B. Before pointing to a location, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select a grid location.

D. After a location has been selected, COSMOS will create a receiving node in that location.

   Note: If this screen location is already filled, COSMOS will display the message:
   create node: this location is already occupied

6.21 Transit

A. After the transit menu option is chosen, COSMOS will display the message:
   create node: point to a grid location

B. Before pointing to a location, the user can pan or zoom the Product Structure Diagram and move or scale any view box.

C. COSMOS now expects the user to select a grid location.

D. After a location has been selected, the Resource Types Menu appears on the data terminal screen. COSMOS will then display the message:
   select a resource type first

6-6
E. COSMOS now expects the user to select a resource type on the data terminal.

F. After a resource type has been selected, COSMOS will display the message:

   select a resource or exit

G. All the resources belonging to this resource type are now displayed on the data terminal. The user must select a resource from the Resource Menu on then data terminal or press PF3 to exit. If a resource is selected, a transit node will be created in the previously specified location.

   Note: If this position is not available, COSMOS will display the message:

   create node: this location is already occupied

6.22 Stock

A. After the stock menu option has been selected, COSMOS will display the message:

   create node: point to a grid location

B. Before pointing to a location, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select a grid location.

D. If the selected location is occupied, COSMOS will display the message:

   create node: this location is already occupied

E. After an unoccupied location is chosen, the Resource Types Menu will appear on the data terminal screen. COSMOS will then display the message:

   select a resource type first

F. After the user selects a resource type, COSMOS will display the message:

   select a resource or exit

G. The Resource Menu appears on the data terminal screen. Select a resource or press PF3 to exit this menu. If a resource is selected, a stock node will be created in the previously specified location.

6.23 Operation

A. After operation has been selected, COSMOS will display the message:

   create node: point to a grid location

B. Before pointing to a location, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.
C. COSMOS now expects the user to select a grid location.

D. After a grid location is selected, an operation node will be created in that location.

   Note: If this screen location is already filled, COSMOS will display the message:

   create node: this location is already occupied

6.24 **Label**

A. After label has been selected, COSMOS will display the message:

   create node: point to a grid location

B. Before pointing to a location, the user can pan or zoom the Product Structure Diagram, and move or scale any view box.

C. COSMOS now expects the user to select a grid location.

D. After a grid location is selected, a label node will be created in that location.

   Note: If this screen location is already filled, COSMOS will display the message:

   create node: this location is already occupied

6.25 **Simulator**

Refer to Chapter 10.

6.26 **Get Case**

A. When Get Case is selected, a Case Selection Menu appears on the data entry terminal screen.

B. The user can either display one of the existing cases, or the user can create a new case.

**To Display an existing case:**

1. Move the cursor to the file name and press ENTER. If the selected case already does not have a view, the Product Structure Diagram of the case will appear as an active view, and Product Structure Diagram menu will become the active menu.

**To create a new case:**

1. Press PF3 to exit the Case selection menu.

2. A new case name is requested on the data entry terminal screen:

   please enter new case name (8 chars maximum) = = >
3. Type in a new case name. Invalid case names include case names which already exist and blank character strings. Invalid case names cause COSMOS to request a new case name. After the user gives a valid case name, a blank Product Structure Diagram view appears at the upper left side of the user area, and the Product Structure Diagram menu becomes the active menu.

6.27 Save

Save the case to the sql files.

6.28 Rename

A. COSMOS will display the message:

   enter new case on data terminal

B. On the data entry terminal, COSMOS will display the message:

   rename: enter new case name = = >

C. If the user presses the ENTER Key, no case will be created. If then the user enters a case name, COSMOS will determine if it is a valid case name. See section 6.26 for notes on valid case names. If the name duplicates an existing case, COSMOS will display:

   rename: this case_name is already used by other case, 
   please type a new case_name

If the case name entered is valid, then all the associated graphs of that case will be given the new case name. This includes the Product Structure Diagram, the Resource Chart and all the Task Charts.

6.29 File

The case of the active view will be saved to the sql flat files, and the graphs will disappear after they are saved.

6.30 Help

Invoke the help menu.

6.31 Exit

Exit to main menu.
Chapter 7
TASK CHART MENU

On the Task Chart menu, the Task Chart view is the active view. The user can make any of the following selections:

<table>
<thead>
<tr>
<th>pan left</th>
<th>view node</th>
<th>move</th>
</tr>
</thead>
<tbody>
<tr>
<td>pan right</td>
<td>goto node</td>
<td>push</td>
</tr>
<tr>
<td>pan up</td>
<td>view edge</td>
<td>pull</td>
</tr>
<tr>
<td>pan down</td>
<td>edit edge</td>
<td>draw</td>
</tr>
<tr>
<td>zoom in</td>
<td>del edge</td>
<td>get case</td>
</tr>
<tr>
<td>zoom out</td>
<td>del node</td>
<td>save</td>
</tr>
<tr>
<td>reset</td>
<td>store</td>
<td>rename</td>
</tr>
<tr>
<td>probable</td>
<td>setup</td>
<td>file</td>
</tr>
<tr>
<td>edit node</td>
<td>process</td>
<td>help</td>
</tr>
<tr>
<td>move node</td>
<td>inspect</td>
<td>exit</td>
</tr>
</tbody>
</table>

The following are the descriptions of all the Task Chart functions:

7.1 **Pan Left**

The Task Chart is moved 20% to the right. This feature displays more of the left side of the Task Chart in the active window.

7.2 **Pan Right**

The Task Chart is moved 20% to the left. This feature displays more of the right side of the Task Chart in the active window.

7.3 **Pan Up**

The Task Chart is moved 20% downward. This feature displays more of the top of the Task Chart in the active window.

7.4 **Pan Down**

The Task Chart is moved 20% upward. This feature displays more of the bottom of the Task Chart in the active window.

7.5 **Zoom In**

The range of the x-axis in the Task Chart is decreased by 20%.
7.6 **Zoom Out**

The range of the x-axis in the Task Chart is increased by 20%.

7.7 **Reset**

The Task Chart is reset back to the original scale.

7.8 **View Node**

A. When **view node** is selected, COSMOS will respond with the message:

```
view node: select a node to view only
```

Here, the user must choose a node to view. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

B. After the user has selected a node, the Task Node Record Menu will be shown on the data entry terminal screen. COSMOS will display the message:

```
view node on the data entry terminal screen
```

See the section of this manual concerning the Task Node Editor.

7.9 **Edit Node**

A. When **edit node** is selected, COSMOS will respond with the message:

```
edit node: select a node to edit
```

Here, the user must choose a node to edit. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

B. After the user has selected a node, the Task Node Record Menu will be shown on the data entry terminal screen. COSMOS will display this message:

```
edit node on the data entry terminal screen
```

See the section of this manual concerning the Task Node Editor.
7.10 **Move Node**

A. When **move node** is selected, COSMOS will respond with the message:

```plaintext
move node: select node to move
```

Here, the user must choose a node to move. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

B. After the user has selected a node to move, COSMOS will display this message:

```plaintext
move node: point to a grid location
```

C. After the user points to a grid location, the selected node will be moved to the new location. All the edges pointing from or into this node will follow the node to the new location.

*Note:* If the new location is already occupied by another node, then a warning message will appear and the user must select another location.

7.11 **Del Node**

A. When **del node** is selected, COSMOS will respond with the message:

```plaintext
delete node: select node to delete
```

Here, the user must choose a node to delete. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

B. After the user selects a node, the node and all the edges pointing from and into this node will disappear. If it is a draw node, then the edge from the inventory node into the operation node will be deleted from the Product Structure Diagram. If it is a store node, then the yield edge which joins the operation node and the inventory node in the Product Structure Diagram will be deleted.

7.12 **Goto Node**

A. When **goto node** is selected, COSMOS will respond with the message:

```plaintext
select a task node on the data entry terminal screen
```

B. All the task nodes of the Task Chart are shown on the data entry terminal screen.

C. The user can press PF3 to exit without making a choice. To make a selection, the user must move the cursor to the desired task node and press ENTER. The Task Chart will be moved so that the selected node is in the center of the window.
7.13 **View Node**

A. When **view node** is selected, COSMOS will respond with the message:

```
view edge: pick edge to view only
```

Before picking an edge to view, the user can pan or zoom the Task Chart, and move or scale any view box.

B. COSMOS now expects the user to select an edge. After an edge has been selected, information about that edge will be shown on the data entry terminal screen. COSMOS will display this message:

```
view edge on the data entry terminal screen
```

C. The user can press PF3 to exit.

7.14 **Edit Edge**

A. When **edit edge** is selected, COSMOS will respond with the message:

```
edit edge: pick edge to change
```

Here, the user must choose an edge to edit. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

B. After an edge has been selected, information about that edge will be shown on the data entry terminal screen. COSMOS will display the message:

```
edit edge on the data entry terminal screen.
```

The user can press the TAB or cursor movement keys to go to different entries on the editing menu and change the record. PF3 exits the editing menu.

7.15 **Del Edge**

A. When **del edge** is selected, COSMOS will respond with the message:

```
delete edge: pick edge to delete
```

Here, the user must choose an edge to delete. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

B. After an edge has been selected, it will be deleted.
7.16 **Draw**

A. When **draw** is selected, COSMOS will respond with the message:

   *create node: point to a grid location*

   Here, the user must choose a grid location. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

   **Note:** If the user selects a location which is already occupied by another node, COSMOS will display the message:

   *create node: this location is already occupied*

B. After an edge has been selected, COSMOS will display the message:

   *select a part_no for this draw node*

   The user must now select a part number on the data entry terminal screen. If the user does not wish to create a draw node, PF3 will exit the menu.

C. If the user selects a part number, COSMOS will determine if an edge already exists between the selected part number and the operation node. If such an edge exist, then a draw node will be created in the task chart. However, an edge will not be created in the Product Structure Diagram. The resource type and resource associated with the newly created draw node will be the same as those of the inventory node. If an edge does not yet exist between the operation and the inventory nodes, both that edge and a draw node will be created. As above, the resource type and resource are the same as the inventory node.

7.17 **Store**

A. When **store** is selected, COSMOS will respond with the message:

   *create node: point to a grid location*

   Here, the user must pick a grid location. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

   **Note:** If the user selects a location which is occupied, COSMOS will display the message:

   *create node: this location is already occupied*

B. After the user points to an unoccupied location, COSMOS will display the message:

   *select a part number for this store node*

   The user can select a part number on the data entry terminal screen. Pressing PF3 will terminate this menu.
C. After the user selects a part number, COSMOS will determine if an edge exists between the operation node owning this Task Chart and the selected part number. If there is such an edge, the store node will be created, but the edge will not be created. Here, the resource type and resource are the same as the inventory node.

If there is no edge between the operation node and the selected part number, COSMOS will create both an edge joining the part number and the store node. Again, the resource type and resource are the same as the inventory node.

7.18 Setup

A. When setup is selected, COSMOS will respond with the message:

create node: point to a grid location

Here, the user must pick a grid location. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

Note: If the user selects a location which is occupied, COSMOS will display the message:

create node: this location is already occupied

B. If the screen position is not occupied, the node will appear on the graphics screen.

C. After an available location is selected, the Resource Types Menu appears on the data entry terminal screen. COSMOS will now prompt the user to select a resource type by displaying the message:

select a resource or exit

The Resource Menu now appears on the data terminal screen. This menu lists all the resources belonging to this resource type. Select a resource or press PF3 to exit this menu. If a resource is selected, a setup node will be created in the selected location.

7.19 Process

A. When process is selected, COSMOS will respond with the message:

create node: point to a grid location

Here, the user must pick a grid location. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

Note: If the user selects a location which is already occupied by an icon, COSMOS will display the message:

create node: this location is already occupied

7-6
B. If an acceptable location has been selected, the Resource Types Menu will appear on the data terminal. COSMOS now expects the user to select a resource type and resource from the menus COSMOS will display on the data terminal. After the user specifies the resource information, a process node will appear.

7.20 Inspect

A. When inspect is selected, COSMOS will respond with the message:

create node: point to a grid location

Here, the user must pick a grid location. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

Note: If the user selects a location which is already occupied by an icon, COSMOS will display the message:

create node: this location is already occupied

B. After an available location is selected, COSMOS will display this message:

select a resource type first

A menu of Resource Types will appear on the data terminal. The user must now select a resource type and press ENTER.

C. COSMOS will now prompt the user to select a resource from the Resource Menu. The Resource Menu displays the resources belonging to this resource type. After the user selects a resource, an inspect node will be created in the previously specified location. Press PF3 to exit without creating the inspect node.

7.21 Move

A. When move is selected, COSMOS will respond with the message:

create node: point to a grid location

Here, the user must pick a grid location. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

Note: If the user selects a location which is already occupied by an icon, COSMOS will display the message:

create node: this location is already occupied

B. After an available location is selected, the Resource Types Menu appears on the data terminal. COSMOS will display the message:

select a resource type first
C. After a resource type is selected, COSMOS will display the message:

    select a resource or exit

The Resource Menu appears on data terminal; this menu lists all the resources belonging to this resource type. The user must select a resource or press PF3 to exit this menu. If a resource is selected, a move node will be created in the previously specified location.

7.22 Push

A. When push is selected, COSMOS will respond with the message:

    create edge: pick from node

Here, the user must pick a from node. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.

Note: If the user selects a location which is already occupied by an icon, COSMOS will display the message:

    create node: this location is already occupied

B. After picking a from node (e.g. chipA), COSMOS will display the message:

    CHIPA selected, pick into node

C. The user must now select an into node. Before selecting the into node, the user can pan or zoom the Task Chart, and move or scale any view box.

Note: If there is an edge joining the selected from node and the selected into node, COSMOS will display the message:

    add edge: the edge is already existing

D. If an edge does not exist between the from (chipA) node and the into node (chipB), COSMOS will display the message:

    CHIPB selected

7.23 Pull

A. When pull is selected, COSMOS will respond with the message:

    create edge: pick from node

Here, the user must pick a from node. Before making a selection, the user can pan or zoom the Task Chart, and move or scale any view.
B. After picking a from node (e.g. chipA), COSMOS will display the message:

CHIPA selected, pick into node

Before picking an into node, the user can pan or zoom the Task Chart, and move or scale any view box.

C. With the graphics mouse, select the into node of the new edge.

D. After selecting an into node (e.g. chipB), COSMOS will check if there is an edge joining the selected from node and the selected into node. If there is an existing edge, COSMOS will display this message:

Add edge: the edge is already existing

If there is no edge already, COSMOS will display this message:

CHIPB selected

7.24 Adding a Probabilistic Edge

A. After probable is highlighted, COSMOS will display the message:

create edge: pick from node

Here, the user must pick a from node. Before picking a from node, the user can pan or zoom the Task Chart and move or scale any view box.

B. After a from node is selected with the graphics mouse (e.g. chipA), COSMOS will display the message:

CHIPA selected, pick into node

Here the user must pick an into node. Before picking a from node, the user can pan or zoom the Task Chart and move or scale any view box.

C. After the user has selected the into node of the new edge, (e.g. chipB) COSMOS will check if there is an edge which joins the from node and into node. If there is such an edge, COSMOS will display the message:

add edge: the edge is already existing

If there is no edge between the two nodes currently defined, COSMOS will display the message:

CHIPB selected

and the new probabilistic edge will appear. The default probability for a probabilistic edge is 1.0.

Note: When the user adds more than one probabilistic edge from a task node, the user must adjust the probabilities of the individual edges using the edit edge menu function to ensure that their sum does not exceed 1.0.
7.25 **Get Case**

Same as `get case` described in Chapter 6.

7.26 **Save**

Same as `save` described in Chapter 6.

7.27 **Rename**

Same as `rename` described in Chapter 6.

7.28 **File**

Same as `file` described in Chapter 6.

7.29 **Help**

Invoke help menu.

7.30 **Exit**

Exit to main menu.
Chapter 8
The Resource Chart Menu

On the Resource Chart menu, the Resource Chart view is the active view. You can make any of the following selections shown on the window area:

<table>
<thead>
<tr>
<th>pan left</th>
<th>zoom out</th>
<th>del node</th>
<th>rename</th>
</tr>
</thead>
<tbody>
<tr>
<td>pan right</td>
<td>reset</td>
<td>add node</td>
<td>file</td>
</tr>
<tr>
<td>pan up</td>
<td>view node</td>
<td>edit rtyp</td>
<td>help</td>
</tr>
<tr>
<td>pan down</td>
<td>edit node</td>
<td>get case</td>
<td>exit</td>
</tr>
<tr>
<td>zoom in</td>
<td>move node</td>
<td>save</td>
<td></td>
</tr>
</tbody>
</table>

The following are the descriptions of all the Resource Chart functions:

8.1 **Pan Left**

The Resource Chart is moved 20% to the right, so that the left part of the Resource Chart is displayed in the active window.

8.2 **Pan Right**

The Resource Chart is moved 20% to the left, so that the right part of the Resource Chart is displayed in the active window.

8.3 **Pan Up**

The Resource Chart is moved 20% downward, so that the top part of the Resource Chart is displayed in the active window.

8.4 **Pan Down**

The Resource Chart is moved 20% upward, so that the bottom part of the Resource Chart is displayed in the active window.

8.5 **Zoom In**

The range of the x-axis in the Resource Chart is decreased by 20%.

8.6 **Zoom Out**

The range of the x-axis in the Resource Chart is decreased by 20%.

8.7 **Reset**

The Resource Chart is reset back to the original scale.
8.8 **View Node**

A. After **view node** is selected, COSMOS will display this message:

```
view node: select a node to view only
```

Before picking a node to view, the user can pan or zoom the Resource Chart, and move or scale any view box.

B. After the user selects a resource node, the information about the selected node will be shown on the data entry terminal screen. See the manual section dealing with editing resource records.

8.9 **Edit Node**

A. After **edit node** is selected, COSMOS will display this message:

```
edit node: select a node to edit
```

Before picking a node to edit, the user can pan or zoom the Resource Chart, and move or scale any view box.

B. After the user selects a resource node, the information about the selected node will be shown on the data entry terminal screen. See the manual section dealing with editing resource records.

8.10 **Move Node**

A. After **move node** is selected, COSMOS will display the message:

```
move node: pick node to move
```

Before picking a node to move, the user can zoom or pan the Resource Chart, and move or scale any view box.

B. After you have selected a node, COSMOS will display the message:

```
move node: point to a grid location
```

C. Point to a new grid location with the graphics mouse and press a selector key. The selected node will be moved to the new location. In addition, all the edges pointing from or into this node will follow this node to the new location.

**Note:** If the new location is already occupied by other node, then a warning message will appear and the user will be requested to select another location.
8.11 Del Node

A. After del node is selected, COSMOS will display this message:

   delete node: pick node to delete

   Before picking a node to delete, the user can zoom or pan the Resource Chart, and move or scale any view.

B. After the user has selected a node, the node will disappear.

8.12 Add Node

A. After add node is selected, COSMOS will display this message:

   point to a grid location

   Before pointing to a grid location, you can zoom or pan the Resource Chart, and move or view box.

B. After the user has selected a grid location, COSMOS will display the message:

   select are source type on data entry terminal screen

C. The Resource Type Selection will then appear on the data entry terminal screen. The user may now select, add or delete any source type at this time.

D. COSMOS will now prompt the user to select a resource type. Move the menu cursor to the resource type and press the ENTER key. A new resource node will appear on the Resource Chart graph with the selected resource type. While in the Resource Type editor, the user may add and delete resource types. See the section of this manual dealing with the Resource Type Editor for further information.

E. PF3 exits this editor.

8.13 Get Case

Same as get case described in Chapter 6.

8.14 Save

Same as save described in Chapter 6.

8.15 Rename

Same as rename described in Chapter 6.
8.16 **File**

Same as file described in Chapter 6.

8.17 **Help**

Help menu.

8.18 **Exit**

Exits to main menu.
Chapter 9
COSMOS DATA EDITORS

9.1 Editing Overview

The COSMOS editing screens enable the user to view and modify the different data components which comprise a COSMOS simulation model. The screens are accessed from the COSMOS graphical editing screens.

By convention, all modifiable fields are highlighted. In order to change a data value, move the data terminal cursor over the selected field (using the TAB or cursor movement keys), and type the new values. Then press ENTER.

While editing a record on the data terminal, the graphics mouse cannot be used to interact with the graphics screen. The PF3 key will return control to the graphics screen and will also terminate the editing process. If other function keys are available, they will be listed at the bottom of the editing screen. For example, the State Transition Record Editor allows the user to scroll to the previous and subsequent records using the PF7 and PF8 keys.

Finally, the user may change several fields of a record by typing the new values into the fields before pressing the ENTER key.

9.2 Operation Node Editor

9.2.1 Screen Function Description

Operation records are associated with the product structure diagram. Each operation owns a task chart containing the sequence of steps that define the operation. Operation records correspond to the SOLOPER disk data type.

9.2.2 Field Descriptions

Name: This is the name associated with this operation node. To change the name, type in the new operation name over the old one. Invalid operation names include blank strings or names which already exist. If there is a duplication of operation names, COSMOS will display the message:

error: PSD name duplicated

Location.X: Graphical X screen coordinate of the operation icon.

Location.Y: Graphical Y screen coordinate of the operation icon. Display only.

Fixed time: This field will be used to control the operation of simulation schedulers to be implemented in the future. Currently, the field is edit-checked for a numeric value.
Variable Time: This field will be used to control the operation of simulation schedulers to be implemented in the future. Currently, the field is edit-checked for a numeric value.

Economic order quantity: This field will be used to control the operation of simulation schedulers to be implemented in the future. Currently, the field is edit-checked for a numeric value.

Material Activity Records: The number displayed shows the number of Material Activity Records associated with this operation. To view/edit the numbers, enter any character in this field.

Figure 13

COSMOS 2.0 OPERATION NODE EDITOR D$EDOPER

Name: RECEIVE_HANDLE

Location : X: -0.6572265625  Y: 0.8349609375
Fixed time : 0.0
Variable Time: 0.0
Economic order quantity: 0.0

Material Activity Records: 1
(Mark with 'X' to edit)

Enter Update Help! PF1 Quit PF3
9.3 **Product Structure Edge Editor**

9.3.1 **Record Type Description**

Product Structure Diagram Edge records define the logical connection between operation and inventory nodes in the product structure diagram. These edges contain information to determine the rate inventory enters and is produced by an operation, as well as operation scheduling priority if more than one edge is defined into an operation. Product Structure Diagram Edge records correspond to the SQLPSDE disk data type.

9.3.2 **Field Descriptions**

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Type</td>
<td>This field describes the type of edge joining two nodes in the product structure diagram. Possible edge types are 'YIELD', linking inventory nodes into operation nodes, and 'BILL OF MATERIALS' linking operation nodes into inventory nodes. The field is not modifiable.</td>
</tr>
<tr>
<td>Units</td>
<td>The units field describes the units which the number represents, e.g. pounds, or pieces. This string is strictly for user convenience and does not designate a conversion factor.</td>
</tr>
<tr>
<td>Quantity</td>
<td>This field is used only by bill of materials edges. (Type = Real)</td>
</tr>
<tr>
<td>Planned yield</td>
<td>This field is used only by yield edges.</td>
</tr>
<tr>
<td>Sourcing</td>
<td>If more than one edge is defined into an operation node, the sourcing determines the processing order of the edges (from lowest to highest). (Type = Integer)</td>
</tr>
<tr>
<td>Transfer coefficient</td>
<td>The transfer coefficient is the multiplier for the number of units. For example, suppose we create a PUSH link where the FROM node is a sheet metal cutter (PROCESS node) and the TO node is the next stage in processing (e.g. an INSPECT node). If the cutter chops one big unit into 10 little units, then the store node must process 10 units for every unit the cutter processes. Here, the transfer coefficient would be 10.</td>
</tr>
<tr>
<td>From Node Information</td>
<td>These fields describe the from node that the given edge joins with the into node. Display only.</td>
</tr>
<tr>
<td>Into Node Information</td>
<td>These fields describe the into node that the given edge joins with the the from node. Display only.</td>
</tr>
</tbody>
</table>
9.4 Material Activity Record Editor

9.4.1 Record Type Description

Material activity records are attached to a task node in the form of a list. The user may scroll through any Material Activity Records defined by using the PF7/PF8 function keys, or press PF3 to exit. This corresponds to the SQLMATA disk data type.

9.4.2 Field Descriptions

Interval

The time interval of each material activity record indicates when the shipment must be made or when the delivery arrives. For the last material activity record on the list, events will be generated periodically using the time interval.

Warning: The model used for the random model in the last material activity record must be 'CONSTANT.'

For material activity records other than the final one, exactly one event will be generated, at the time given as the 'interval.'
Quantity

The quantity in the material activity record indicates the number of units which are shipped or received. This is a random variable consisting of a probability model and 1-3 numeric parameters.

Activity type:

This field should always be set to BURST.

Swap Record ordering:

This field allows the user to change the ordering of material activity records. Enter a 'P' in this field to exchange the list position of the current record with that of the material activity record directly preceding it. Enter an 'F' to switch the order of the current material activity record with the one directly following it. If there is no previous or next record, then no ordering change is performed.

Figure 15

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>MATERIAL ACTIVITY EDITOR</th>
<th>D$EDMATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record no: 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Interval length:
- MODEL : CONSTANT
- MEAN : 1.8

Quantity:
- MODEL : CONSTANT
- MEAN : 1

Activity type: BURST

Swap Record ordering:
- P: Current <--> Previous
- F: Current <--> Following

9.5 Inventory Record Editor

9.5.1 Record Type Description

Inventory records are associated with the product structure diagram. There is exactly one inventory record for each part defined in a model. Inventory records provide information concerning inventory levels and policy assumptions. These levels and assumptions ultimately affect the rates and levels at which parts are drawn and replenished. This corresponds to the SQLINV disk data type.
9.5.2 Field Descriptions

Part number: This field uniquely identifies this part to COSMOS. To change the part number, type in a new part number over the old one. This field is required and may contain a blank string. If the user specifies a part name which already exists, COSMOS will display the message:

**error: PSD name duplicated**

Location.X: Graphical X screen coordinate of the operation icon. Display only.

Location.Y: Graphical Y screen coordinate of the operation icon. Display only.

Unit: Arbitrary text set by user (e.g. pieces, gallons, etc.). This field is optional, and no edit checking is performed.

Cost: (Type = Real)

On hand: This is the quantity of this part currently available. (Type = Real)

On order: This field represents the sum of amounts of this part stored in this resource throughout the test case. (Type = Real)

On request: (Type = Real)

Item type: Valid entries are STOCK and TRANSIT. Changing this field will invoke the Measure Type editor, described elsewhere in this manual.

Resource Address.Category: This is the resource category of the Resource Type displayed below. Output only.

Resource Address.Type: Type any character(s) on the Resource Type line. When COSMOS displays the Resource Type Selection Menu, select one of the resource types displayed. After making a selection, the user may optionally select a resource from the Resource Menu.

Resource Address.Zone: If the user changes this field, the Resource Selection Menu will appear. The Resource selection menu will list all the valid resources for the resource address type.

Policy.Draw point: COSMOS edit-checks this field for numeric values, but this is not used in the current COSMOS implementation.

Policy.Draw down quantity: COSMOS edit-checks this field for numeric values, but this is not used in the current COSMOS implementation.

Policy.Delivery date: (Type = Real)

Policy.Reorder point: This field contains the inventory level at which inventory is restocked. (Type = Real)
Policy.Reorder interval: (Type = Real)

Policy.Reorder quantity: When an order for inventory is triggered, this field is the amount of inventory associated with each request. (Type = Real)

Figure 16

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>INVENTORY RECORD</th>
<th>D$EDINVN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part number: POW1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location: X: -0.5419921875 Y: 0.7998046875</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit :</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost : 0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On hand : 0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On order : 0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>On request : 0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item type : STOCK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Policy: Default Resource Address:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draw point : 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Draw down quantity : 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivery date : 0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorder point : -78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorder interval: : 2.147483647E+08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reorder quantity: : 78</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.6 Product Structure Diagram Label Editor

9.6.1 Record Type Description

Product structure diagram labels are used for identification purposes.

9.6.2 Field Descriptions

Title: This is a required informational line of text displayed on the graphics terminal. The default value: is "****** USER LABEL ******" (Type = Alphanumeric)

Line1: This is an optional description line. (Type = Alphanumeric)

Line2: This is an optional description line. (Type = Alphanumeric)

Line3: This is an optional description line. (Type = Alphanumeric)
9.7 Task Node Editor

The Task Node editor is accessible from the Task Chart screen by highlighting the edit node menu item and then selecting a task node icon. On the graphics terminal, COSMOS will display the message:

edit node on the data entry terminal screen

Information relating to this task node will appear on the data entry terminal. Press the TAB or cursor movement keys to move the cursor to the field to be changed. To return to the Task Node editing screen, press the PF3 function key.

9.7.1 Record Type Description

Task records correspond to the SQLTSK1 disk data type.

9.7.2 Field Descriptions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>This is the name of the task node. It may not be blank or duplicate an existing name within the task chart.</td>
</tr>
<tr>
<td>Task Chart:</td>
<td>This is the name of the task chart owning this task node. Display only.</td>
</tr>
<tr>
<td>Type:</td>
<td>Task node type (e.g. SETUP, MOVE, STORE, etc.). Display only.</td>
</tr>
<tr>
<td>Priority:</td>
<td>(Type = Integer)</td>
</tr>
<tr>
<td>Process Batch Min/Max</td>
<td>This is the maximum quantity allowed for any batch with this task node. If a larger quantity is requested, then it will be broken up into larger batches. (Type = Real)</td>
</tr>
<tr>
<td>Time per:</td>
<td>This field describes whether the time field (described below) is calculated on a per unit or per batch basis. The field is alphanumeric and modifiable. If the user changes this field, COSMOS will prompt the user to choose a new value from a selection menu. Possible values are PERBATCH or PERUNIT.</td>
</tr>
</tbody>
</table>
Time
This a random variable consisting of an alphanumeric model type, followed by 1-3 decimal fields, the number depending upon the random model type used. Each field appropriate to this model will have a label describing its purpose; those not pertaining to this model are unlabeled and unmodifiable.

To change the type of the random variable, type any character into the type field to invoke the random variable type editor.

To modify any of the numeric parameters associated with this random model, type a new numeric value into the desired field(s).

Size per unit
This field describes the storage capacity of a resource consumed by one unit of this part.

Resulting state
This field applies to setup nodes only. It describes to state a resource will be left in when the setup operation is performed, typically READY. No edit checking is performed on this field. (Type = Alphanumeric)

Primary Requirements
These fields define the task size and storage size which must be met by the corresponding task capacity and storage capacity of the resource designated by the Primary Resource Address. (Type = Real)

Primary Resource Address
These fields indicate the resource or class of resources which must be allocated in order to process material through the task node.

If the user changes the resource type, COSMOS will invoke the resource type selection menu a list of currently defined resource types for the model.

After the user has selected a resource type, COSMOS will display a list of resources belonging to the resource type specified. The user may optionally select a resource by moving the cursor to the name of the desired resource and pressing ENTER. PF3 cancels this option.

To change only the Primary Resource, type any character in the field marked RESOURCE and press ENTER. Type any characters on the type line and press ENTER. COSMOS will display a list of resources of the specified type. The user may optionally select a resource by moving the cursor to the name of the desired resource and pressing ENTER.

Secondary Requirements:
Secondary resource requests are the same as primary resource requests except that:

A. No space is required or used on the primary resource (because the 'stuff' is on the PRIMARY resource).

B. There may be zero, one or more secondary requests.

C. There are no state transition records.
Secondary Requirements exist in the form of a list belonging to the current task node. The number of Secondary Requirement records contained in this list is displayed in this field. For example, a '2' in this field means the list contains two records. To view or edit this list of Secondary Requirement records, enter any character in this field and press ENTER.

If the Secondary Resource Requirements list is empty, COSMOS will display this message:

This list is empty, Add a new node? (Y/N)

If the user enters 'N', COSMOS will immediately return to the Task Node Editing screen without adding a new Resource Requirement record.

If the user enters 'Y', the Resource Type Selection Menu will appear. Select a resource type and press the ENTER key. The Resource Selection Menu now appears listing all the resources in this resource type. Select a resource and press ENTER. The Resource Requirements Editor screen now appears and displays the newly created record.

**Edit Draw Part#:**
(Draw nodes only)

Type any characters on the Draw Part# line and press ENTER. The Part Number Selection Menu will appear. Move the cursor to a part number and press ENTER. In the Product Structure Diagram view, the edge joining the original part number and the operation, shipping, or receiving node will disappear, and a new edge joining the new part number and the operation, shipping, or receiving node will be created.

**Please Note:** The operation, shipping, or receiving node owns the Task Chart which this task node is in.

**Edit Store Part#:**
(Store nodes only)

Type any characters on the Store Part# line and press ENTER. The Part Number Selection Menu will appear. Move the cursor to a part number and press ENTER. In the PSD view, the edge joining the original part number and the operation, shipping, or receiving node will disappear, and a new edge joining the new part number and the operation, shipping, or receiving node will be created.

**Please Note:** The operation, shipping, or receiving node owns the Task Chart which this task node is in.
This field applies to setup task nodes only. The number displayed represents the number of state transformation records in the list associated with this setup task node. To access this list, type any character in the State Transformation Records line and press ENTER. If the state transformation list is empty, then COSMOS will display this message:

This list is empty, Add a new node? (Y/N)

If the user enters 'N', COSMOS will return to the Task Node Editing Screen without adding a new State Transformation record. If the user enters 'Y', COSMOS will create and display a State Transformation record containing default values.

These fields apply to move nodes only. To change the information displayed, type in new data and press ENTER. To change the Move Type value, type any characters on the Move Type line and press ENTER. COSMOS will display the current value of this field, followed by a menu of valid Move Type values. Enter the number of the desired value and press ENTER.

Figure 18

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>TASK NODE EDITOR</th>
<th>D$EDTSKN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: OPN79</td>
<td>Task Chart: EXCHANGE 2</td>
<td></td>
</tr>
<tr>
<td>Type: SETUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Priority: 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Batch: Min: 0.0 Max: 1</td>
<td>Time per: PERBATCH</td>
<td></td>
</tr>
<tr>
<td>Model: CONSTANT</td>
<td>MEAN: 1</td>
<td></td>
</tr>
<tr>
<td>Task amount: 1</td>
<td>Size per unit: 1</td>
<td>Resulting state: READY</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary requirements:</td>
<td>Task size: 1</td>
<td>Storage size: 1</td>
</tr>
<tr>
<td>Resource Address: Name: PASTE-PLANT</td>
<td>State transformation records:</td>
<td></td>
</tr>
<tr>
<td>Type: PASTE-PLANT</td>
<td></td>
<td>Category: MACHINE</td>
</tr>
<tr>
<td>Secondary Requirements: 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9.8 State Transition Record Editor

The user may access the State Transition editor by selecting the State Transition option in the Resource Type Editor. If the state transformation list is not empty, then the State Transformation Records Menu will appear. If the list is empty, the user will be asked if a state transition record, containing default values, should be added. Answering 'N' to the prompt will return the user to the Task Node Editor.

After accessing the State Transition Editor, the user may change the information displayed by typing over it and pressing the ENTER key.

The following function keys are available:

PF8 - Scroll to the next node.
PF7 - Scroll to the previous node.
PF10 - Add a new State Transformation record.
PF3 - Return to Resource Type editing screen

9.8.1 Record Type Description

State Transition records correspond to the SQLSTTR disk data type.

9.8.2 Field Descriptions

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource type:</td>
<td>This field identifies the resource type 'owning' this state transition list. It is not modifiable.</td>
</tr>
<tr>
<td>Resulting state:</td>
<td>Arbitrary characters representing the state of the resource after the state transition is executed.</td>
</tr>
<tr>
<td>Starting state:</td>
<td>Arbitrary characters.</td>
</tr>
<tr>
<td>Time to setup</td>
<td>Random variable specifying how long it takes to change a resource from the starting state to the resulting state.</td>
</tr>
</tbody>
</table>

Figure 19

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>STATE TRANSITION EDITOR</th>
<th>D$EDSTTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record of Resource type:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resulting state: READY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Starting state : READY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to setup:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEL</td>
<td>CONSTANT</td>
<td></td>
</tr>
<tr>
<td>MEAN</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>(param2)</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>(param3)</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
9.9 Task Link Editor

9.9.1 Record Type Description

Task Link records define the logical connection of task nodes in task charts. The user may edit information affecting the rate of inventory flow from one operation to another. This corresponds to the SQLINK disk data type.

9.9.2 Field Descriptions

Unit: Arbitrary text set by users (e.g. pieces, gallons, etc.).
Transfer size:
Transfer coefficient:
Probability: Used for probabilistic edges only. Represents the probability that a certain edge will be chosen. May have the range 0.0-1.0, where 1.0 = 100%. (Type = Real)

Figure 20

```
COSMOS 2.0 TASK LINK RECORD EDITOR D$TSKLNK

Unit: Transfer size: 1
Transfer coefficient: 1
Probability: 0.0
```

9.10 Resource Editor

The resource editor allows the user to edit information associated with the resources defined in a test case. Resources belong to a resource type, and the resource type in turn belongs to one of five resource categories. The user may change any one of the highlighted fields.

To edit a resource node, press the TAB or cursor movement keys to move the cursor around the data entry terminal screen, type the new data over the old entry, then press RETURN.

9.10.1 Record Type Description

Resource records contain information concerning the resources available to a model. Resource records correspond to the SQLRES disk data type.
9.10.2 Field Descriptions

Name: To add a resource name, type in a new name in an unoccupied location on the menu screen. If the name does not already exist, it will appear on the menu screen.

Resource Category: To change the resource category of a resource type, the user must select the edit typ menu item on the main COSMOS graphic menu.

Resource type: To change the type of a resource, type any characters on the Resource Type Line, and press RETURN. The Resource Type menu will appear. Move the cursor to any resource type and press RETURN.

Name: This is the name associated with this resource. This name may not duplicate a name already defined for a particular resource type.

Resource Category: This is the category of the resource type which owns this resource. The categories provided are meant to correspond to conceptual divisions among the resources, but do not affect the logical operation of the simulation. Display only.

Resource type: To change this field, type any character over the entry displayed. COSMOS will then display a menu of valid resource types.

Screen location.X: X screen coordinate of the resource icon.

Screen location.Y: Y screen coordinate of the resource icon.

Grid location: Physical location of this resource on the factory floor.

State: Arbitrary string only used for Setup Task nodes.

Tasking Used: A measure of how busy a resource is. (Type = Real)

Space Used: A measure of how full a resource is. (Type = Real)

Maintenance: Currently unused. It will be used to describe the necessary maintenance for this resource.

Repair State: Current repair state of this resource. After the user types a character into this field, COSMOS will display a menu of repair states to select from. Current values: Failed or Operational.

Time since service: This field displays the time that this resource was last serviced. (Type = Real)

Failure time: If a resource has failed, this field contains the time that the failure occurred. (Type = Real)
Contents Records: This field contains the number of contents records associated with this resource. This corresponds to the SQLCONT disk data type.

To edit a contents record, type 'X' on the Contents Records line, and press RETURN. The Content Node Editor will appear on the data entry terminal screen.

**Figure 21**

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>RESOURCE EDITOR</th>
<th>D$EDRESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: CON-BAKED2</td>
<td>Resource Category: TRACK</td>
<td></td>
</tr>
<tr>
<td>Resource type : CON-BAKED2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Screen location : X: 0.48571428571429  Y: -0.0642857142857</td>
<td>Grid location : X: 0.25714285714286  Y: -0.3214285714285</td>
<td></td>
</tr>
<tr>
<td>State: Utilization : 0.0</td>
<td>Tasking Used: 0.0</td>
<td></td>
</tr>
<tr>
<td>Space Used : 0.0</td>
<td>Maintenance: Repair State: OPERATIONAL</td>
<td></td>
</tr>
<tr>
<td>Time since service: 0.0</td>
<td>Failure time: 2.147483647E+08</td>
<td></td>
</tr>
<tr>
<td>Contents Records: 1</td>
<td>(Enter any character to view)</td>
<td></td>
</tr>
</tbody>
</table>

### 9.11 Contents Node Editor

Press PF8 to go to the next content node.
Press PF7 to go to the previous content node.
Press RETURN or PF3 to exit.

#### 9.11.1 Record Type Description

#### 9.11.2 Field Descriptions

- **Resource Address.Category:** Resource category of the resource which owns this Contents node. Display only.
- **Resource Address.Type:** Resource category of the resource which owns this Contents node. Display only.
- **Resource Address.Name:** Resource category of the resource which owns this Contents node. Display only.
Part number: To change the contents part number, type any characters on the part number line and press RETURN. The Part Number Selection Menu will appear. Move the cursor to a part number and press RETURN.

Quantity: The number of units of this part contained in this resource.

Space Used: The amount of space occupied in this resource by these parts.

**Figure 22**

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>CONTENT NODE EDITOR</th>
<th>D$EDCONT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Address:</td>
<td>Category: TRACK</td>
<td></td>
</tr>
<tr>
<td>Type : CON-BAKED2</td>
<td>Name : CON-BAKED2</td>
<td></td>
</tr>
<tr>
<td>Part number: CL-BAK</td>
<td>Quantity : 30</td>
<td></td>
</tr>
<tr>
<td>Space Used : 0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**9.12 Resource Type Selection Screen**

The Resource Type Selection enables the user to select and modify resource types associated with a COSMOS model. In some instances, this screen will only permit the selection of a resource. The current options will be displayed on the message area of the display screen.

**9.12.1 Record Type Description**

Resource type records contain information concerning resources associated with each type. This includes the resources' time and frequency of failure, as well as their task and storage capacities. Resource types always belong to one of the predefined COSMOS resource categories. Resource type records correspond to the SQLRTYP disk data type.

This screen is accessible by selecting the highlight `edit rtyp` menu item on the Main Menu on the graphics terminal.

1. To select a resource type, move the cursor to a resource type, and press ENTER.
2. To add resource type, move the cursor to a category and type in a new resource type. If the name of the new resource type is duplicated within the resource type, COSMOS will display the message:

   error: duplicate name

If the resource name entered was valid, the new resource will appear on the data terminal.

3. To delete a resource type, move the cursor to a resource type and enter a blank string into the field or erase it using the ERASE EOF key, then press the ENTER key. COSMOS will display this warning:

   Resources associated with resource type(s) to be deleted will also be deleted.
   Do you want to continue? (Y/N)

If the user enters 'Y', then all the resources belonging to this resource type will be deleted. 'N' will immediately cancel the requested deletion and return to the Resource Type Editing screen.

---

Figure 23

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>RESOURCE TYPE SELECTION MENU</th>
<th>D$SLRTYP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACHINE</td>
<td>Select, add, or delete a resource type</td>
<td></td>
</tr>
<tr>
<td>PASTE-PLANT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRANE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLEAN-MACH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELECTROLYSIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXCHANGE 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXCHANGE 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARTIF-ELEC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARTIF-TR1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARTIF-TR2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUXCRA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHIP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| STORAGE      |                             |          |
| PYR-GREEN    |                             |          |
| PYR-BAKED    |                             |          |
| OVEN         |                             |          |
| CONT-ELEC    |                             |          |
| CONT-TR1     |                             |          |
| CONT-TR2     |                             |          |
| ALPHA        |                             |          |
| BETA         |                             |          |
| DUMMY        |                             |          |
| INTERM       |                             |          |
| RODDING      |                             |          |
| ARTPOW1      |                             |          |

| FIXTURE      |                             |          |
9.13 **Resource Type Editing Screen**

To change the data in a field, type new data into that field, and press enter. The name of a resource type may also be changed in this way, but the new resource name may not duplicate one which already exists. If the user selects a name which has been previously defined, COSMOS will display the message:

`error: duplicate name`

If the resource type name is valid, the new resource type name will appear on the data terminal.

To move a resource type to another category, type any characters over the old one, and press ENTER. COSMOS will prompt the user to select a new category from a menu. Enter the number of the new category and press ENTER.

To add a new resource type, move the cursor to a category and type in a new resource type. If the name of the new resource type is duplicated, COSMOS will display the message:

`error: duplicate name`

If the name is not duplicated, the name will appear on the Resource Type menu.

9.13.1 **Field Descriptions**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Character string uniquely identifying this resource type. Valid definitions exclude blank strings and other resource type within this category.</td>
</tr>
<tr>
<td>Category:</td>
<td>This field contains the resource category of this resource type (e.g. Storage, Labor). To change this value, type any character into this field, and COSMOS will display a menu of valid resource categories.</td>
</tr>
<tr>
<td>Task unit:</td>
<td>Character string. Optional.</td>
</tr>
<tr>
<td>Task capacity.Min:</td>
<td>(Type = Real)</td>
</tr>
<tr>
<td>Task capacity.Max:</td>
<td>(Type = Real)</td>
</tr>
<tr>
<td>Storage unit:</td>
<td>Character string identifying the units the storage capacity is measured in. Optional.</td>
</tr>
<tr>
<td>Storage capacity.Min:</td>
<td></td>
</tr>
<tr>
<td>Storage capacity.Max:</td>
<td></td>
</tr>
<tr>
<td>Time to repair</td>
<td>Random variable representing the amount of time required to repair a resource of this type.</td>
</tr>
<tr>
<td>Time to fail</td>
<td>Random variable representing the amount of time between failures for a resource of this type.</td>
</tr>
</tbody>
</table>
**Figure 24**

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>Resource Type Editor</th>
<th>D$EDRTYP</th>
</tr>
</thead>
</table>

**Name:** TRAIN-1  
**Category:** CARRIER  
**Task unit:**  
**Task capacity:** Min: 0.0  Max: 1  
**Storage unit:**  
**Storage capacity:** Min: 0.0  Max: 7.698798E+07

**Time to fail:**  
MODEL: CONSTANT  
MEAN: 2.147483647E+08  
(param2): 0.0  
(param3): 0.0

**Time to repair:**  
MODEL: CONSTANT  
MEAN: 0.0  
(param2): 0.0  
(param3): 0.0
Chapter 10
THE COSMOS SIMULATOR

10.1 Background

The COSMOS simulator consists of some "schedulers" and the simulator itself. The user selects a scheduler from the menu, and that scheduler, in turn, manages the simulator. Below are field descriptions. The contents of these fields determine several aspects of the simulation.

10.2 Field Descriptions

Time Now (Display only) Show the progress of the simulation. The simulation will stop when this time reaches the value specified in the End Time field.

Events Now (Display only) Show how many events the simulator has processed so far.

End Time Specify the number of "time units" the simulation is to run.

End Events Specify the number of events the simulation is to run.

Run Controls Specify the setting (on/off) of the indicated run control.

The run controls include:

| Backorder | Storage Free | Task Free |
| Setup Free | Inspect Free | Failure Free |
| Access Free | Shrink Lot | Lost Sales |
| Verbose |

Output Data Files Specify the filename which is to hold the results of the simulation. Several files of varying types with the given name will be created.

Policy File Name (Optional) Specify a filename which contains policy change records to be incorporated into the simulation.

Release Lot File (Optional) Specify a filename which contains lot release records to be incorporated into the simulation.

Scheduler Specify which of the shown scheduling rules are to be used to run the "factory" during the simulation. When a lot release file is used, the typical choice here is "None Selectd."

Burn-in Time Specify the amount of simulator "time units" that will elapse before the first simulation results will be written to the output files.

Random Seed Specify the seed to use in random number generation. This may be used to vary the simulation results.
Comments: Specify some comments which will uniquely identify this particular simulation run.

PF1 Help!: Display a help message. This feature is not yet implemented.

PF2 Reset: Clear out the simulator data structures, and prepare for a "fresh" run.

PF3 Quit: Return to COSMOS from the simulator.

PF6 Undo Chgs: Undo the most recent round of changes the fields of the simulator screen.

PF9 Browse: Walk through the data structure built by the simulator when it last ran. This is used primarily for debugging your test case, (e.g. This feature can help answer such questions as "Why didn't any of 'part X' get produced?")

PF12 Run: Begin the simulation. This function is unavailable until the minimum number of fields necessary for simulation have been filled in.

---

COSMOS Simulator Screen

<table>
<thead>
<tr>
<th>COSMOS 2.0</th>
<th>Simulation Run Controls</th>
<th>S$MENU00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Now:</td>
<td>0.0</td>
<td>Events Now:</td>
</tr>
</tbody>
</table>

- Backorder
- Storage Free
- Task Free
- Setup Free
- Inspect Free
- Failure Free
- Access Free
- Shrink Lot
- Lost Sales

- Burn-in
  Time: | 0.0 |

Legend:
> Option Selected
- Option Not Selected

Comments: |

PF1 Help!  PF2 Reset  PF3 Quit  PF6 Undo Chgs  PF9 Browse  PF12 Run

10-2