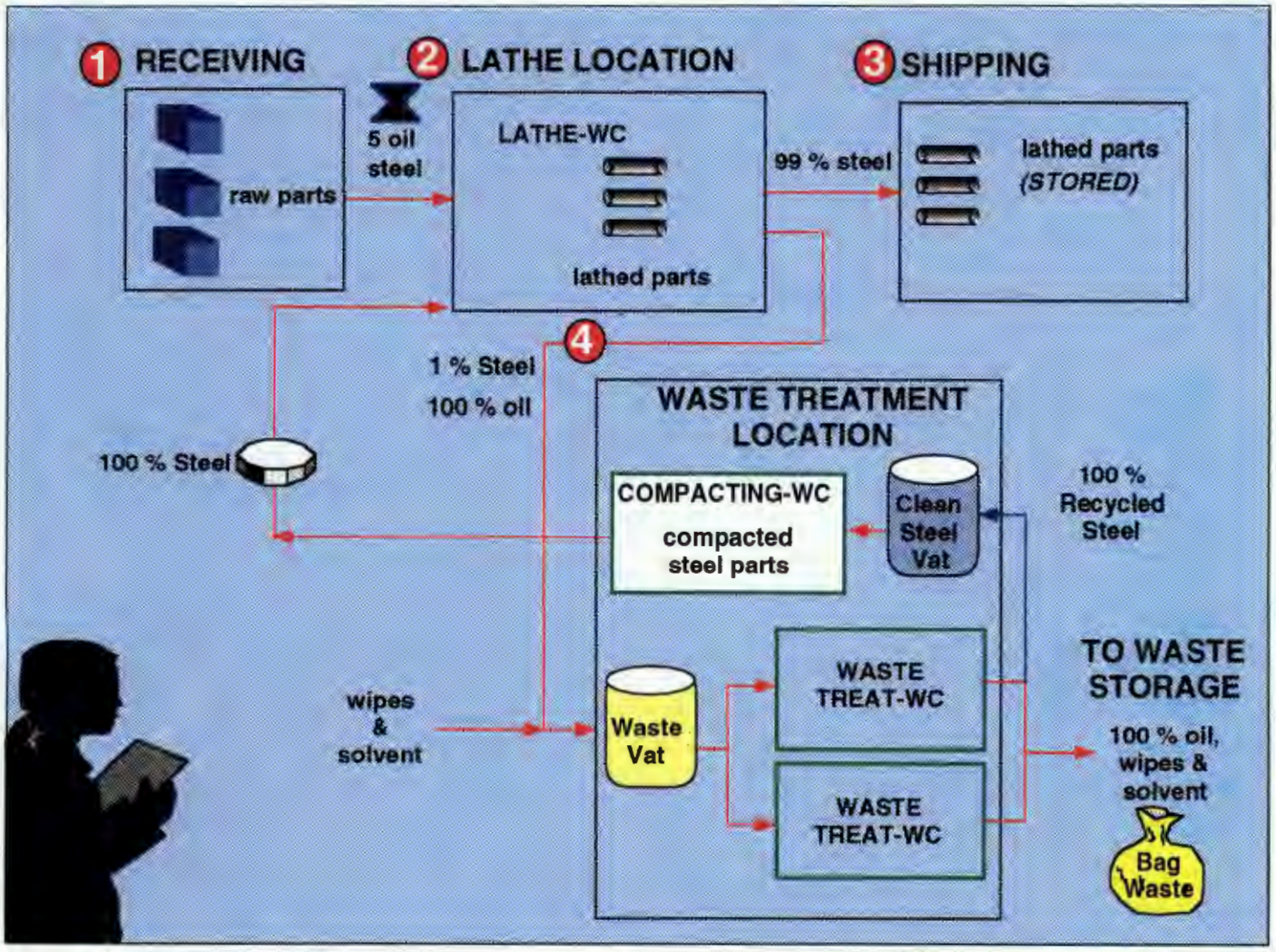
LIBRARY COPY



INTRODUCTION TO PROCESS WASTE ASSESSMENT SYSTEM FOR WASTE COORDINATORS

TSA-7 TECHNOLOGY MODELING AND ANALYSIS

Los Alamos National Laboratory

Technology and Safety Assessment

13766

INTRODUCTION TO THE PROCESS WASTE ASSESSMENT SYSTEM

(PWA)

For Waste Management Coordinators

Presented by
TSA-7 Technology Modeling & Analysis
Training by
T&D

INTRODUCTION TO THE PROCESS WASTE ASSESSMENT SYSTEM

(PWA)

For Waste Management Coordinators

Presented by

TS A-7 Technology Modeling & Analysis

Training by

~~HRD-3~~
T&D

WASTE MINIMIZATION PROGRAM

Thanks to a Lab-designed computer modeling system, waste coordinators are learning to manipulate up to several hundred material and process variables to generate the least amount of waste possible. The system, called the Process Waste Assessment System, is easy to use and provides almost instant feedback on changes that affect waste generation, according to Donna Smith with Technology Modeling and Analysis (A-7).

"With this modeling system you can almost endlessly change process and materials variables to determine under what conditions you can generate less waste. Not only can you use it for waste minimization, but you also can place limits on processes, such as the maximum amount of material an area can hold, and find out under what conditions they're achievable," said Smith.

The Lab and other government facilities must submit to the Department of Energy a plan to assess waste-generating processes to identify and help implement waste minimization options, said Jeff Weinrach, the Process Waste Assessment System project leader with Environmental Management's Division Office (EM-DO). Other programs model processes, but not waste generation, too. A collaboration between A-7 and EM resulted in the novel computer software.

"It works by keeping track of what happens at every step of a process," said Smith. "A simple example would be a tooling operation. Let's say 20 tons of steel a day flow into the operation and 15 tons of finished parts flow out. Our system tracks what happens to those other five tons. It might be four tons wind up as shavings that can be recycled and one ton of steel that ends up as rejected parts that are recycled in a different way.

"But, it doesn't end there. The machines also use oil and coolants. All of that can be tracked and accounted for. It even allows for material transformation. If you start out with a known amount of hydrogen and oxygen, you will end up with an expected amount of water, so you always know how much of something is going in, or coming out of, the system," said Smith.

Reports can be generated that provide an accurate picture of waste generation and can provide information for such things as material substitution. Material substitution, replacing one material with another that performs better or is less hazardous, is easier when you can easily see every instance in which that material is being used and in what quantities, Smith said.

The ease in learning the system is one of its strong points, said MaryAna Eames, an instructor with Employee/Organization Development (HRD-3).

"The system is user-friendly. It's not difficult to learn and the reaction we've had to the training has been positive. An advantage is not only is it a good tool, but it's fun to use in the sense of 'Let's see what happens if we do this' and then get an immediate response. You also don't need complex equipment to run it. It runs with popular personal computers," said Eames.

"The software also can be linked to other software systems including INFOCAD that is being used for facility management," said Weinrach. "We're working toward the point at which someone could pull up a room on a blueprint-type display and look at all the processes and materials there. While each area would model its own processes, information could be collected to get an accurate picture of all similar processes at the Lab. That would help in identifying more Labwide waste minimization opportunities."

The reports generated not only would help minimize waste and improve processes, but could come in handy for environmental audits. Rather than having to search for information, a significant amount could be pulled from the system, he said.

The potential of the system to accurately model processes and waste streams already is being recognized, said Smith. It is being used at the Rocky Flats nuclear plant for waste processing and treatment, at Pantex for weapon disassembly and in industry to look at all the production and waste treatment processes within a printed, wiring-board manufacturing plant.

The market potential for software could be large as more facilities face the need to minimize waste and understand how processes affect waste, said Weinrach. Commercialization of the software is being planned.

Anyone interested in taking the free, three-day Process Waste Assessment class should contact Eames at 7-3211.

—Linda Anderman

Instructors

Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, New Mexico 87545

Kathy Burris
MS-F609
505-665-1394
burris@mimsy.lanl.gov

MaryAna Eames, HRD-3
MS-M589
505-667-3211

Harry Egdorf
MS-F609
505-667-3507
hwe@lanl.gov

Steve Painter
MS-F609
505-667-5856
swp@lanl.gov

Donna Smith
MS-F609
505-667-9473
smith@mimsy.lanl.gov

Jeffrey Weinrach, EM-DO
MS-K572
505-665-8292

1.0. Overview **1**
1.1. Performance-based Learning Objectives: 1
1.2. The Role of a Simulation Models in Waste Management and Minimization 2
1.2.1. Computer -based Simulation Models 3
1.2.2. PWA as a Discrete-event Simulation Model 3
1.3. What is the Process Waste Assessment System and Where Does It Apply? 4
1.4. Where has the Process Modeling System Been Used? 5
1.5. Current Hardware and Software Platforms 6
1.6. Support and Maintenance 6
2.0. Process Modeling System Description **7**
2.1. Material 7
2.1.1. Parts 7
2.1.2. Batches 7
2.1.3. Continuous Streams 7
2.1.4. Material Resources 7
2.2. Material Movement 7
2.3. Material Processing 8
2.4. Reusable Resources 8
2.5. Exposures 8
2.6. Shifts 8
2.7. Breakdowns and Failures 9
2.8. Processing Outputs 9
2.9. Data Output 9
2.10. How The Process Waste Assessment System Works 10
2.10.1. Part Processing 10
2.10.2. Bulk Material Processing 12
2.10.3. Stream Movement Processing 14
2.11. HOW PWA IS STRUCTURED 16
2.11.1. Event Architecture 16
3.0. Using the Process Waste Assessment System **18**
3.1. Syntax 18
3.2. The Mechanisms of Object-Oriented Programming Available to the User 19
3.3. Keywords of PWA 22
3.3.1. Times and Dates 24
3.3.2. The Use of Comments in Scenario Files 25
4.0. Materials and Material Containers **26**
4.1. Materials 26
4.2. Material Representations 27
4.3. Parts 28
4.4. Bulk Materials 30
4.5. Streams 31
4.6. Resources 32
4.6.1. Critieria for Identification of Resources 32
4.6.2. Unconstrained Resources 32
4.6.3. Constrained Resources 34
4.6.4. Materials 36

5.0. Reusable Resources 37
5.1. Critieria for Identification of Reusable Resources 37
5.2. Working Shifts 38
5.2.1. Basic Shift Attributes 38
5.3. Operators 39
5.3.1. Basic Operator Attributes 39
6.0. Exposures 41
6.1. Basic Exposure Attributes 41
6.2. Location-Time-Exposed-Operators 42
6.3. Workcenter-Time-Exposed-Operators 42
6.4. Contact-Exposed-Operators 43
6.5. Non-Contact-Exposed-Operators 43
7.0. Locations 44
7.1. Critieria for Identification of Locations 44
7.2. Unconstrained Locations 45
7.2.1. Basic Location Attributes 45
7.2.2. Basic Location Capabilities 48
7.3. Constrained Locations 49
7.3.1. Unit-Capacity-Constrained-Locations 49
7.3.2. Capacity-Constrained-Locations 51
7.3.3. Material-Constrained-Locations 52
7.3.4. Maximal-Footprint-Locations 53
7.3.5. Mixing Types 54
8.0. Workcenters 55
8.1. Critieria for Identification of Workcenters 55
8.2. Basic Workcenter Attributes 56
8.3. The Process-Step-Descriptions Attribute 58
8.4. Descriptions of Process Steps 59
8.5. Initial Status 60
8.6. Inputs 61
8.6.1. Parts 62
8.6.2. Bulk Containers 63
8.6.3. Streams 65
8.7. Material Resources 66
8.8. Reusable Resources 69
8.9. Normal, Failure, and Breakdown Outputs 70
8.10. Outputs 71
8.10.1. Existing Part Output 72
8.10.2. New Part Creation 72
8.10.3. Continuous Stream Creation 73
8.10.4. Bulk Container Output 74
8.10.5. Resupply of the Stock Room 75
8.10.6. Material Balances 76
8.10.7. Absolute Amounts in a Material Balance 77
8.10.8. Material Fractions in a Material Balance 77
8.10.9. Defaulting Rules 78

8.10.10. The Default Material Rule 78
8.10.11. The Default Output Item Rule 79
8.10.12. Material Transformations 80
8.11. Process-Step-Descriptions Examples 82
9.0. Processes 85
9.1. Critieria for Identification of Processes 85
9.2. Process Attributes 86
10.0. Workorders 88
11.0. Random Deviates 89
11.1. Exponential-Random-Deviate 89
11.2. Uniform-Random-Deviate 89
11.3. Normal-Random-Deviate 90
11.4. Constant-Random-Deviate 90
12.0. Running the Simulation 91
12.1. SUN 91
12.1.1. Interactive 91
12.2. Batch Execution 95
12.2.1. Simple Local Batch 95
12.2.2. Batch With Programmatic Control 96
13.0. Utilities for Debugging Scenarios 98
14.0. Auditors 99
14.1. Audited-Locations 99
14.2. Audited-Processes 99
14.3. Audited-Operators 100
14.4. Audited-Resources 100
14.5. Audited-Workcenters 100
14.6. Audited-Parts 101
15.0. Foremen 102
15.1. Quitting-Foremen 102
15.2. Warning-Foremen 103
15.3. Debugging-Foremen 103
15.4. Auditing-Foremen 103
15.5. Reporter-Foremen 104
15.6. Equipment-Foremen 106
15.7. Mixed Foremen 106

Class Introduction

1.0 Overview

1.1 Performance-based Learning Objectives:

Terminal

When the class is completed, the student will possess the required set of skills to:

1. Identify when modelling and simulation can best meet their needs.
2. Identify when the Process Waste Assessment system is an appropriate tool for simulation.
3. Design, run and test a simulation model(s) representative of real-world operations in his/her technical facility.
4. Perform analysis, run reports and interpret the results.

Enabling

1. Describe the purpose and utility of viewing real-world operations in terms of material flows and transformations and how these can be analyzed through the Process Waste Assessment system.
2. Understand the benefits and drawbacks of applying PWA to a real-world processing environment.
3. Develop, test and run scenario input files to construct simulations using PWA to test the efficiencies of such methods in the real-world.
4. Gain proficiency in applying text-driven syntax commands and menu-driven operations to manipulate PWA to perform user-designated processing simulations.
5. Gain proficiency in system operations such as the debugging of errors in text files and the use of customizing in the menu-driven mode.
6. Learn how to produce and interpret reports and perform analysis using PWA.

1.2 The Role of Simulation in Waste Management and Minimization

A simulation is the imitation of the operation of a real-world process or system over time. Whether done by hand or on a computer, a simulation provides you the opportunity to test and measure the outcome of changing, altering or introducing new processes and resources to a real-world operation in your workplace.

For example, in the case of Waste Management Coordinators, if your job was to monitor leaks within a radioactive liquid waste pipeline in your facility; you could employ simulation and a simulation model to characterize or imitate your real-world environment (set up a scenario) and then conveniently test the outcome of changing or adding one or several variables such as: number of metering devices per linear foot of pipeline, number of leakage detectors, number of personnel added to monitor the pipeline and times when checks should take place. After you set up a scenario or "model" of your real-world environment, you can use simulation to test the application of these various factors and resources. If your goal was to have no more than two leaks a year, the simulation will provide you with the optimal mix of variables and resources to achieve your goal. In other words, the simulation will help you find the optimal number of metering devices, leakage detectors, personnel and the monitoring times you should employ to achieve your goal of 2 or less leaks per year.

Simulation and simulation models are employed in almost every type of industry and various types of simulation models are used. Simulation and the use of simulation models affords people efficient, practical and economic ways to test and measure how they might improve their current operations to achieve goals.

Some of the advantages of simulation models are:

1) Once a model is constructed, it can be used to analyze proposed design, policiy or other operational changes.

2) Simulations are easier to use than analytical models and can means of deriving a solution to a problem.be used to analyze a proposed system requiring minimal input data.

3) Simulation models are economical and easily adaptable to any environment.

4) Simulation models generate data that can be used to estimate any conceivable performance measure.

5) In some instances, simulation is the only means of deriving a solution to a problem.

Can you think of three operations in your work environment that may be improved through analysis using a simulation and modelling tool?

How would you model these operations?

1.2.1 Computer -based Simulation Models

Many problems are simple enough to be solved by mathematical methods such as differential calculus, probability theory, algebraic methods or other mathematical techniques. The solution usually consists of one or more numerical parameters which are called measures of performance of the system. However, many real-word systems are too complex for mathematical solutions. In these instances, numerical, computer-based simulation can be used to imitate the behavior of the system over time.

1.2.2 PWA as a Discrete-event Simulation Model

The Process Modeling System may also be considered as an extensible collection of objects or set of libraries. An analyst uses this library to define a system in terminology common to the analyst's problem domain. A programmer can extend the library by way of published protocols to produce a tool specialized to one particular processing facility and its particular requirements.

1.3 What is the Process Waste Assessment System and Where Does It Apply?

Operation of a facility is a complex dynamic process. Managing and monitoring the generation and flow of waste through a facility can be overwhelming. Since the science of waste management and waste minimization is very new, there are few tools available to assist in managing and monitoring the generation, treatment and disposal of waste. Generally, organizations must understand how the facility which generates the waste operates and how in-house equipment stores, destroys and/or disposes of the waste. Such understanding often requires tools to examine complex dynamic processes.

Many important questions may be addressed by examining the processing of material as it moves through a facility. Such questions include:

1) How much product material can be produced in the next six months or year using current equipment?

2) How much and what types of waste will be generated over the next year with current equipment and schedules?

3) How much and what types of waste will be eliminated by a proposed process change? Can production schedules be met with the proposed change?

4) Where does the material input to a process go?

5) How much of each material used in a facility goes to solid waste, liquid waste, gaseous waste , recycling and reprocessing?

Examination of two aspects of material processing has proven useful in the analysis of waste management facilities and waste minimization activities:

1. Examination of the movement of material from place to place in a facility, and in particular, the constraints on the movement of material, allows an analyst to address issues of procedures, sizes, and capacities of the facility.
2. Examination of the transformation of the material by pieces of equipment in the facility allows an analyst to examine issues of equipment types and numbers, process efficiencies, and ancillary issues related to procedures, worker requirements, waste and by-product generation, product throughput, and resource availability.

In theory, such questions would be best addressed by examining the actual facility, changing the facility and determining the effects of the changes. In practice, however, it often is not practical or is too expensive to perform experiments on the actual facility. This is why we employ a simulation tool such as PWA to give us the answers -- cheaply and quickly without the need to alter our facility, work-schedule or resources.

The Process Waste Assessment System is a simulation tool designed to allow an analyst to describe a material processing facility in terms of the material movement from place to place and the material transformation by pieces of equipment.

The modelling and management of processes can be facilitated through an understanding of several key concepts:

-Know how to identify the materials in your real-world system(s).

-Know how to specify the containers of material

-Know how to identify the physical locations and equipment in the real world that are important in material movement transformations.

Once you have learned the above, you will have the skills to apply the key concept:

-Know what elements in your real-world system should be put into a model. Once you can do this, you will be ready to experience the real benefits of the Process Waste Assessment system (PWA).

In the Process Waste Assessment System, material movement may be:

1. constrained or unconstrained, as well as
2. in discrete parts, batches, or continuous flows.

Material processing may be:

1. in discrete parts or
2. in batches of varying size, and also
3. constrained or unconstrained by the availability of materials, personnel, and time.

1.4 Where has the Process Modeling System Been Used?

The PWA system is based on the Process Modelling System developed at Los Alamos National Laboratory by A-7, Technoogy, Modelling & Analysis. The Process Modelling System has been the foundation for a number of different applications and projects:

- Pantex: Weapon returns and disassembly, and the staging and shipping of nuclear material.
- Ecocircuit: All production and waste treatment processes within a printed wiring board manufacturing plant including chemical, mechanical, and assembly processes.
- Rocky Flats: Processing and waste treatment of plutonium in an existing facility including chemical, foundry, mechanical, and assembly processes.
- TASP: Processing and waste treatment of plutonium in a future facility with technology trade-off studies, including chemical, mechanical, and assembly processes.
- ARIES: Processes in an experimental plutonium processing line.
- Russian Plutonium Storage: Movement and handling of nuclear materials in a future storage facility.

- Process Waste Assessment: All waste-generating processes within Los Alamos National Laboratory to identify opportunities for process improvements and quantify benefits from proposed process changes.
- Organization 9000: Sandia National Laboratory. Application classified.

1.5 Current Hardware and Software Platforms

- Sun Sparc
  SunOS 4.1
  Lucid Common Lisp 4.0
- Apple Macintosh
  System 7
  Apple MCL 2.0
- IBM/PC
  MS-DOS 5.0
  Windows 3.1
  Franz inc., Allegro CL/PC

1.6 Support and Maintenance

- Bug Reports
- Software Distribution
- Training

2.0 Process Waste Assessment System Description

2.1 Material

PWA simulates the movement and processing of material, which may be contained in discrete parts, batches of material in bulk containers, continuous streams and/or stocks of material resources.

2.1.1 Parts

Parts move through a facility from one location to another. The material in the parts may be processed by a workcenter in the location. Users define the paths that parts may take as they are transferred through a facility and the processing that may occur at any location within the facility.

2.1.2 Batches

Batches of material are held in bulk containers. All or part of the material in a bulk container may be removed for processing by a workcenter.

2.1.3 Continuous Streams

A stream of material moves from one location to another, where it is immediately processed by a workcenter. Streams are used to simulate continuous processing such as may be found in a refinery.

2.1.4 Material Resources

Resources are inputs that a workcenter needs to process material. A resource may be defined such that

1. its availability constrains production, or
2. its availability does not constrain production, but the amount used is monitored.

Material resources are incorporated into a workcenter's output and usually represent materials obtained from a central supply or stockroom.

2.2 Material Movement

Material can move from one location to another location and be processed. Material movement may be constrained or unconstrained. That is, material may or may not be able to move from one location to another because of certain constraints.

Movement of material may be constrained by any or all of the following:

1. There may be insufficient space available in a location to receive an incoming part.

2. There may be a limit on the number of parts that can be present in a location at any one time.
3. There may be a restriction on the types and amounts of material that can be present in a location at any one time.
4. There may be a limit on the size of the largest part that can be in a location.

Triggering mechanisms can be set so that when levels of material or numbers of parts are reached, other activities within the model may begin. This is useful when a storage facility becomes full or a feeding facility becomes empty.

2.3 Material Processing

Material can be processed or transformed in workcenters, which represent equipment in a location. Workcenters take input material from parts, batches, or streams and use material resources to produce outputs. Processing may split, consolidate, or transform materials.

Each process performed by a workcenter must maintain a mass balance; that is all input materials must be accounted for in the output materials. Any number of material outputs of any type may be generated. Outputs may be routed to different locations for further processing or storage.

One processing step may generate many products, residues, and wastes, which must all be accounted.

2.4 Reusable Resources

Reusable resources are not incorporated into a workcenter's outputs, but are required for the workcenter to operate. Such resources may be defined so that

1. their availability constrains production, or
2. their availability does not constrain production, but the amount used is monitored.

Examples of reusable resources include operators, and tools.

2.5 Exposures

Operators can accrue exposures as a result of time spent in a location, time spent at a workcenter, and/or contact with exposure-generating materials. Exposure types are user-defined; common radiation and noise are typical examples.

2.6 Shifts

Users may define shift schedules for operators and assign different numbers of operators to different shifts. Schedules indicate the days of the week and hours of the day worked. Schedules may also be defined for the manufacturing plant as a whole and not tied to a specific type of operators.

Users may also define holidays, which are days that the shift might normally be available but will not be available.

2.7 Breakdowns and Failures

Workcenters can experience breakdowns, which are represented as stochastic processes. A breakdown may require that the workcenter be repaired, a subsequent process that may require resources and generate wastes.

Workcenters may fail to produce products of acceptable quality, also according to stochastic processes. Such failures do not require a repair of the workcenter, but may generate outputs that are different from those generated during normal operations.

Several distributions are available to represent how often breakdowns and failures occur, breakdown repair times, and the proportions of outputs that are considered failures.

2.8 Processing Outputs

Waste and residue generation, handling, treatment, disposal, and their effects on the production facility may be modeled in an integrated fashion within PWA. The effects of production on waste treatment and of waste treatment options on production may be considered within an integrated system.

2.9 Data Output

Data collection is available for each object modeled by the simulation. Several report formats may be selected for displaying output information.

2.10 How The Process Waste Assessment System Works

2.10.1 Part Processing

The movement and processing of parts is illustrated in FIGURE 1.

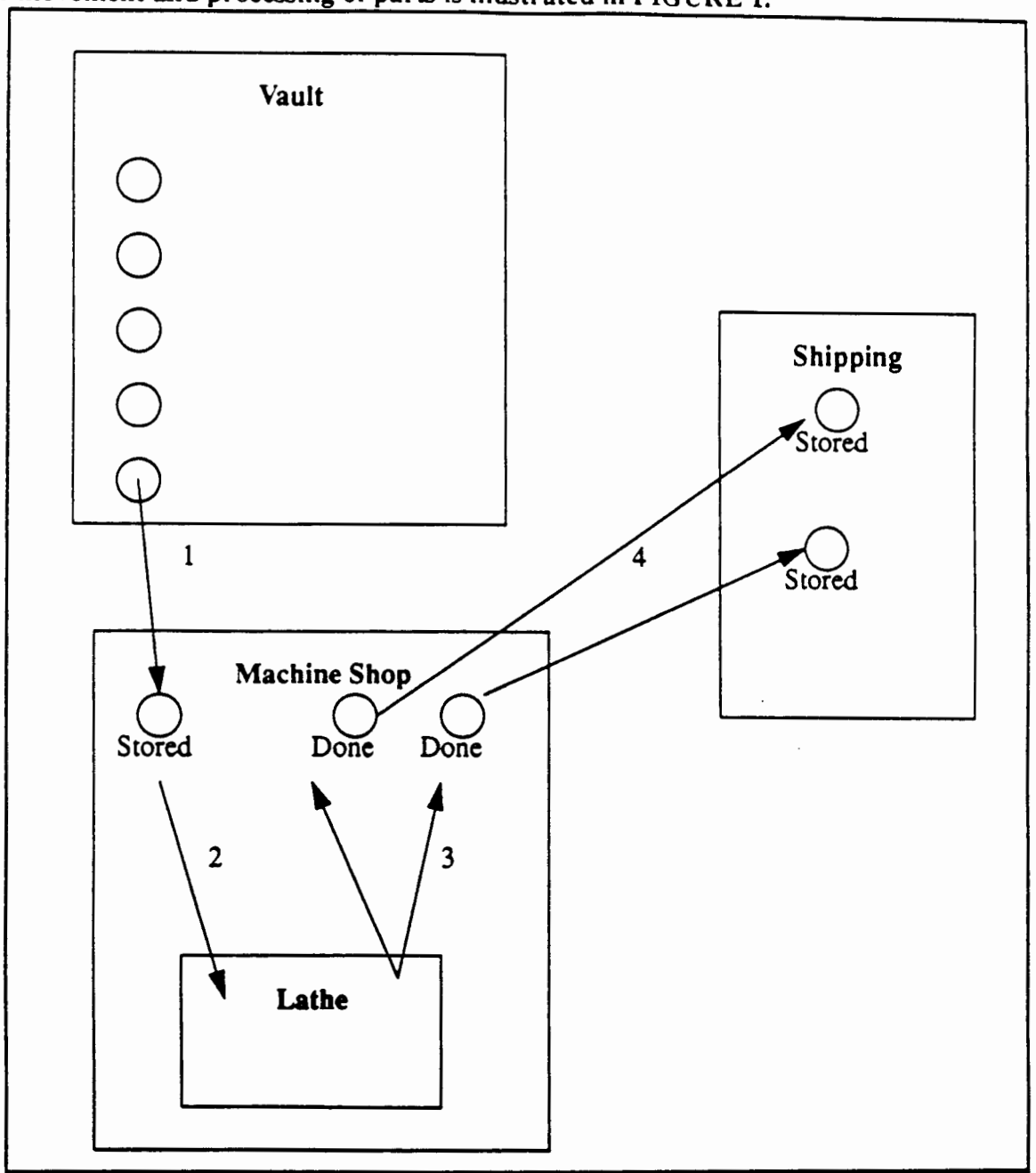


FIGURE 1.

Part Processing and Movement

The steps shown in FIGURE 1. are:

1. A part is moved from the initial location, *vault*, to a new location, *machine shop*. The status of the part is set to **stored**.
2. The part is selected by the *lathe* and processed. Any resources needed are added to the process by the workcenter.
3. The ***lathe workcenter*** finishes processing the part and outputs two parts. The part status of each output part is set to **done**.
4. The parts are moved to the next location, *shipping*.

2.10.2 Bulk Material Processing

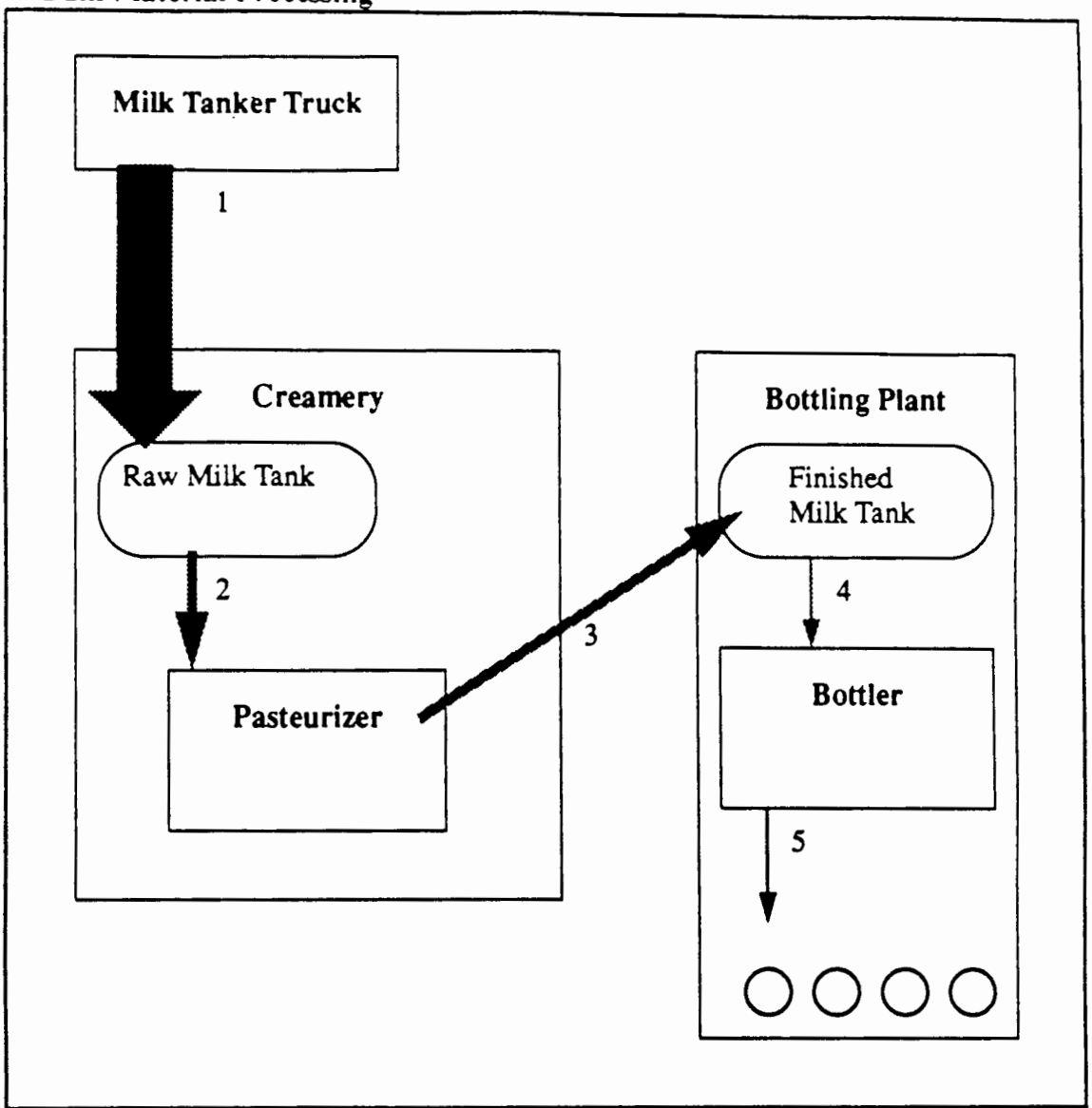


FIGURE 2.
Bulk Material Movement and Processing

The steps shown in FIGURE 2. are:

The movement and processing of bulk materials is illustrated in FIGURE 2. The steps shown in the diagram are:

1. A ***milk tanker truck*** arrives at the creamery and unloads a tank load of raw milk from the dairies into the *raw milk tank.*
2. The ***pasteurizer*** draws a batch of milk from the *raw milk tank.* The batch may be all or part of the milk in the tank.

3. The ***pasteurizer*** puts the processed batch of milk into the *finished-milk tank* at the *bottling plant*.

4. The ***bottler*** draws a batch of milk from the *finished-milk tank*.

5. The ***bottler*** outputs gallon jugs of milk.

2.10.3 Stream Movement Processing

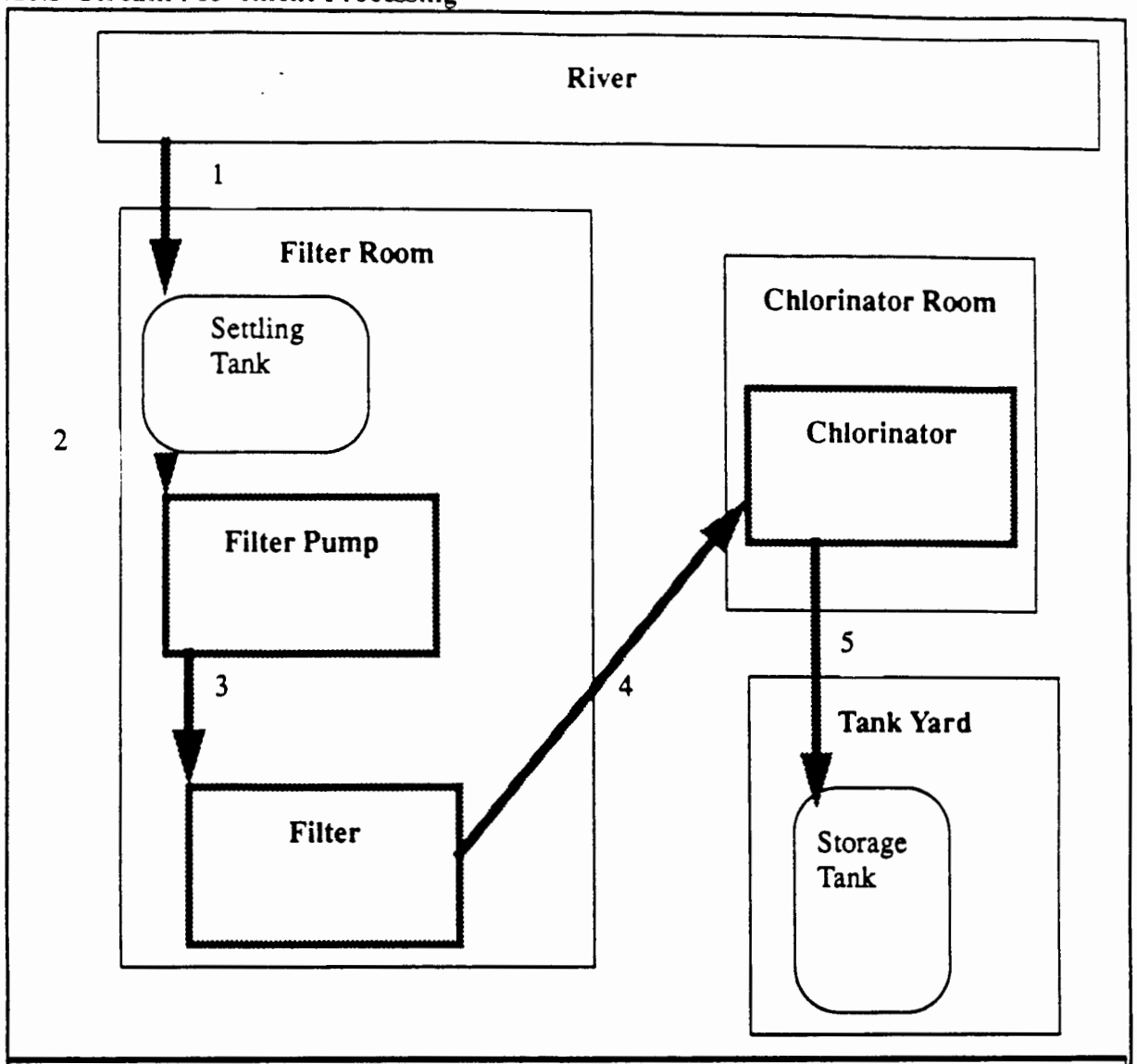


FIGURE 3.
Stream Movement and Processing

The steps shown in FIGURE 3. are:

The movement and processing of streams is illustrated in FIGURE 3. The steps shown in the diagram are:

1. Water is pumped from the *river* into a ***settling tank.***
2. The ***filter pump*** draws water from the *settling tank.*
3. As soon as the ***filter pump*** outputs water, it is immediately processed by the *filter.*

4. As soon as the ***filter*** outputs water, the water moves to the ***chlorinator***, where chlorine gas is introduced.

5. As soon as the ***chlorinator*** outputs water, the water moves to the storage tank yard, where it is held for later use.

2.11 HOW PWA IS STRUCTURED

2.11.1 Event Architecture

PWA is an object-oriented discrete-event simulation tool. The event architecture is structured so that the collection of variables necessary to describe the system at any time -- change only at discrete sets of points in time. This is the inverse of a continuous system in which the variables change continuously over time. PWA, as a discrete-event simulation tool describes the causal relationships between events. Events may be viewed as actions or decision points for the various objects within the simulation. The primary chain of events within PWA is as follows:

1. The state of a location changes.
2. The location assesses its situation, since there was a change in state:
   a. Parts with a **done** status are checked to see if they need to move to another location. If so, the receiving location is notified that a part needs to move.
   b. Each workcenter within the location is given the opportunity to check its state and decide if it should begin processing. Processing can begin if
      i. there are parts with a **stored** status in the location,
      ii. the **stored** parts need to have a process (as defined by the attached workorder) that the workcenter can perform, and,
      iii. the workcenter is in the correct status to begin processing. Workcenters that are busy cannot begin a new process until the first one is finished.
   c. The transfer of parts into the location is started if possible. The transfer is not possible if any constraints for the location would be violated. Constraints include limits on the number of parts or the amount of material allowed in the location.
3. The location posts an event to begin the transfer of any parts into the location. An event to end the transfer is posted for the time when the part will reach the location.
4. Any workcenters in the location that are able to begin processing post events to begin processing.
5. When a workcenter begins processing an end-processing event is posted for the time the processing will be completed.
6. When a workcenter finishes processing, outputs are generated and sent to the appropriate places.
   a. Parts are marked **done** so that the location may decide if they should move.
   b. Materials are placed in bulk containers either within the location or at another location. Bulk containers do not leave the location to which they are assigned.
   c. Streams are directed to destination locations when the locations and workcenters agree on the transfers.
7. Every location that has a change of state will post a decide operation as in Step 2. State changes may be the result of

a. being notified that a part should move,

b. receiving material in a bulk container, or

c. receiving a discrete part or stream of material

This primary chain of events is used by all locations and workcenters to cause decisions and interactions between objects within the model.

3.0 Using the Process Waste Assessment System

The steps for setting up a model of your real-world system is as follows:

1) Decide what in your work environment needs to be modelled, what goals you wish to measure or achieve and if simulation is the appropriate tool to provide you with the results you seek.

2) Identify the objects (locations, workcenters, resources, constraints and other factors) that are present in the system you want to model and assign them names. e.g. Waste-Treatment-Vat, Waste-Treatment-Location, machine-oil, etc.

3) Define the specific characteristics or *"attributes"* that describe each type and sub-type of object. For example, steel-parts have the materials iron and nickel while aluminum parts have the material attribute aluminum.

4 Enter your description of the real-world system into the PWA system and perform your analysis.

++

Using PWA:

The next step in using PWA is to take the data that you developed as outlined above and enter it into the PWA simulation model. The data is entered into the model utilizing a file, or set of files which are called **scenario file(s).** Scenario files must be written in a specific format utilizing pre-defined syntax and a combination of user-defined and system-defined key words. All information describing the facility and its operations are presented to the PWA as a **scenario.**

The scenario files describe two major facets of the real world system being modeled. These are:

- the relationship of objects to each other (such as several places being locations of one type while several other places are locations of some other type differing, perhaps, in the initial values of some attributes, or the existence of some characteristics,)
- The values of attributes of either specific objects or of classes of related objects,

Ancillary information necessary for running the simulation such as starting and ending dates and times for a modeling run, and initial events to be posted are also defined in the scenario.

3.1 Syntax

A simulation is controlled by a scenario which the user defines. In defining a scenario, the user may group and create objects for the model in a number of different ways. Since the PWA is implemented as an object-oriented discrete event simulation, the user may organize the scenarios based on the following mechanisms commonly used in object-oriented programming.

3.2 The Mechanisms of Object-Oriented Programming Available to the User.

Some important mechanisms from the object-oriented programming approach can be employed by users to facilitate the development of scenario files for input into PWA.

1. Group all objects which have common characteristics into a class. (e.g. The class of vehicles).

2. Group all objects which alter the common characteristics of the class and define new common characteristics into appropriate subclasses. (e.g. The subclass of wheeled vehicles)

3) Further group objects belonging to subclasses or classes into" instances" of classes or subclasses which define a specific object in the world that "is a" member of the specified subclass or class. (e.g. Ford Taurus car)

The above rules for classification of objects for use in the PWA model, can be illustrated with an example from the every-day world with which most people are familiar -- inheritance and class structure. The inheritance and class structure for the object vehicles is illustrated in Figure 1. In the example, all vehicles are contained in the highest level class (also called Parent class) VEHICLES. The top level class, the Parent class, will ensure that all vehicles have an engine and the default type of that engine will be gasoline. The example also shows that there are several different types of vehicles which are called sub-classes. The subclasses are WHEELED-VEHICLES, TRACKED-VEHICLES, CARS and TRUCKS. Each subclass consists of related objects which have features or "attributes" which distinguish them further from the parent class and/or contain additional information or redefined values which were inherited from the higher level classes. For example, MY-NEIGHBOR'S-CAR is a specific instance of the class CARS. An instance is a particular single object which is one occurrence of the class. Thus MY-NEIGHBOR'S-CAR is one particular car which inherits all its characteristics from the higher level classes. An instance may set the value of any attribute, but does not add any new attributes.

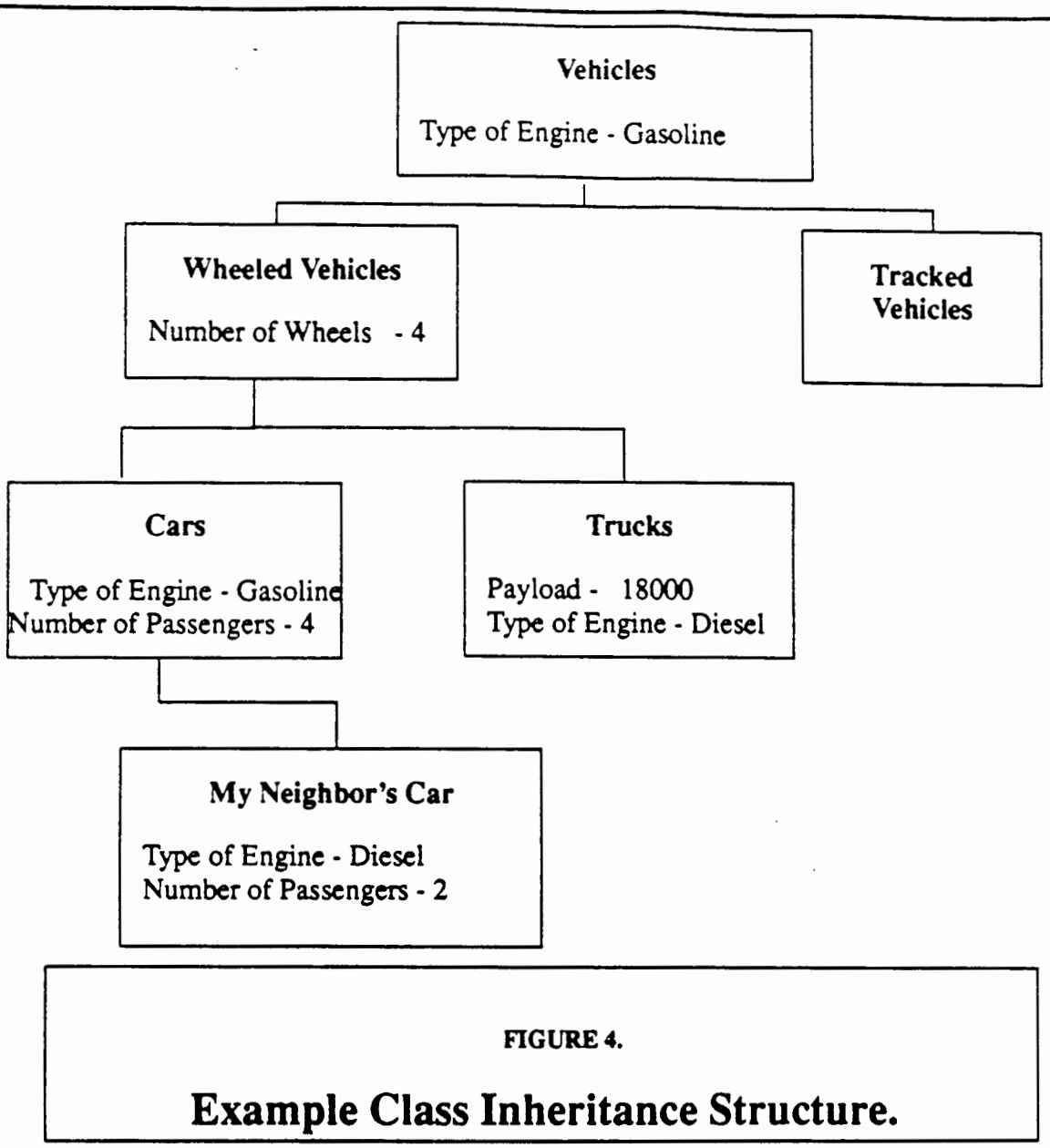


FIGURE 4.

Example Class Inheritance Structure.

There is no single way to define the objects in PWA, however, some class, subclass and instance organizations will generate smaller, more maintainable scenarios. Users should adopt a style and naming convention to assist with consistency within a scenario.

One organizational style is as follows:

Identify a class, define each instance of the class and list all of the attributes describing each instance:

```
:class my-class-1
    :instance building-1
        attribute-1
        attribute-2
    :instance building-2
        attribute-1
        attribute-2
```

Utilizing this structure will create two new objects (instances) of type my-class-1 named building-1 and building-2, each with two attributes. If both building-1 and building-2 have the same attributes, an alternative with identical results would be:

```
:class my-class-1
        attribute-1
        attribute-2
    :instance building-1
    :instance building-2
```

PRE-DEFINED CLASSES:

In order to facilitate writing scenarios and to add consistency to the model, programmers have created a number of different classes which analysts may use to define their real-world facility. Users' may not change the attributes of the programmer defined classes. If the user wants to change attributes, then a subclass of the programmer defined class can be made, and the attributes altered for the user-created subclass.

3.3 Keywords of PWA

Writing scenarios is facilitated for the user by the availability of keywords which indicate the relationships between objects and other important information. PWA users should use the following keywords in their scenario files:

:class

indicates that a new class (also called Parent class) is being defined. The new class will become the **current** class.

:class steel-mill
:class PWA-Materials

:subclass

indicates a new subclass of the defined parent class. When writing the syntax, the user must insert a machinized-subclass to indicate that the subclass definition is complete. The new class will become the current class. The previous current class is remembered so that it can be restored beyond-subclass.

:subclass steel-mills-in-pittsburgh

:end-subclass

indicates a subclass definition is complete. Must have a matching:subclass. The current class will be changed to whatever class was current when the:subclass matching this:end-subclass was read.

:end-subclass steel-mills-in-pittsburgh

:instance

indicates that a specific object should be created which is one instance of the class or subclass currently being defined. This instance will be the **current** instance until either a new instance is defined, or until the current class is changed.

:instance steel-mil-on-grant-ave.

:parent
:parents

indicates that the current class or subclass definition is of some type. The parent-child relationship of a class and its parents represents an **is-a** relationship. This usually is indicated after euclase in order to indicate the type of the class.

:Class my-locations
:Parents (locations)

:include

indicates that another scenario file is to be read at this point. At the end of the indicated file, processing of this scenario description will continue.

```
:include location-file
```

:print

indicates that the related text should be printed on the user's standard output at this time.

```
:print "HELLO"
```

:title

indicates the title you assign to your scenario.

```
:title "WASTE-MANAGEMENT-SCENARIO-1"
```

:clock

indicates the user-specified time at which the simulation is to begin. This must be set before any events are created in the scenario file.

```
:clock (0.0 0 0 1 1 1992)
```

:limit

indicates the time beyond which events will not be processed. Events posted for this time and before will be processed. This is used to give an ending time to the simulation.

```
:limit (0.0 0 0 1 1 1993)
```

:at

indicates that an event will be posted for execution. Some instance must be current for this to be accepted

```
:at (0.0 0 0 1 1 1993) decide-operation
```

<anything else>

indicates that some attribute's value is to be initialized. This initialization will set a default value for a class if no instance is current. If some instance is current, the value will be initialized for that instance only.

```
part-status done
```

Much of the rest of this document describes the various attribute names for the existing types of objects in the PWA and the associated values that can be used.

3.3.1 Times and Dates

The PWA model represents activities that occur in a facility over time. Consequently, there are a number of mechanisms to describe times and dates.

The time and date is written as:

(seconds minutes hours day month year) or,

current-simulation-clock.

Current-simulation-clock is the user specified time the simulation will start.

Intervals of time is expressed in the form:

(seconds minutes hours days).

Note that months and years are not part of the interval definition. This is because three months is a meaningless interval by itself since there are a varying number of days in each month.

A number of keywords are also available to define an interval or span of time. These are:

:daily
:weekly
:first-of-monthly
:end-of-monthly
:bi-monthly
:quarterly
:yearly-calendar
:yearly-fiscal

The dates and intervals are used throughout the model to define when specific events should occur and how often some events should occur, e.g. **Audit-period :daily**

3.3.2 The Use of Comments in Scenario Files

Most good programs are documented and clarified by the use of comments. In PWA, it is a good idea to write descriptive comments about your scenario(s) to better clarify what is taking place and to better communicate the information to others.IT IS RECOMMENDED THAT COMMENTS WITHIN A SCENARIO BE PRECEDED BY 2 SEMICOLONS (;;). This programming convention tells LISP that they are not part of your scenario and should not be processed. You can add as many comments as you like.generated by a semi-colon. All information on the line after the semi-colon is assumed to be a comment.

Example

```
:class warehouse

    unit-capacity 10        ; ; there can only be 10 parts in a warehouse at a time
                              ; ; there are two warehouses at the facility

    :instance building-1
    :instance building-2
```

4.0 Materials and Material Containers

4.1 Materials

PWA models the constraints on the movement of materials through a processing facility, and the splitting, merging, and transformation of sets of materials as they move through a processing facility.

These materials are represented as pairs of the form:

(material-name material-amount)

The material name is the arbitrary name you assign to the material(s) being used in your scenario. These names might be an element such as Fe or Pu, a molecule such as O2 or FeCl3, or a more composite name such as Rust or Filtrate. PWA will place no meaning on such a name. The system has no inherent knowledge of any properties that might be related to these names in the real world (such as an atomic weight associated with the element Fe).

The material amount is a numeric quantity that represents the amount of the material in some units such as kgs, or tons (for mass) or liters (for volume). There are no units associated with this material amount by the simulation system. It is the responsibility of the analyst to ensure that consistency is maintained throughout the model. For example, the analyst should use a unit such as kg consistently, and not intermix kg and g.

These materials are organized into sets of materials represented as lists of the form:

((m1 a1) (m2 a2)...)

These sets of materials will be contained in objects that have special characteristics related to the movement and processing of their contained set of materials. These containers will be described later in this document. For now, it is sufficient to think in terms of the processing of a set of materials such as ((Fe 200) (Ni 23.5) (Co 12.25) (O2 4.33) (Crud 27.652)). Note that each material is enclosed in its own set of parenthesis -- because they represent subsets of the entire group of materials which defines the set. The entire set is then enclosed in a set of parenthesis.

The PWA model operates by moving these sets of materials from place to place. At each place where the set of materials move (called locations), pieces of equipment (workcenters) attached to that location can merge, split, and transform the sets of materials into other sets of materials.At the same time, the simulation model continuously maintains a mass balance such that all materials at the start of such a processing step will be present at the end of the processing step. At the end of the processing step, the newly-produced sets of materials may move on to other locations for additional processing.

Analysis is performed by watching how the sets of materials move and what processing is performed on those sets of materials.

4.2 Material Representations

The basis for operation of the PWA model is sets of material moving around in a processing facility, being split, merged, and transformed. In order to allow these sets of materials to be handled in a consistent way in different situations, each set of materials is held in a material container. Material containers may move from place to place in the simulation model. The material containers provide different ways of accessing the set of materials they contain.

The operation of the simulation models constructed with PWA will be described relative to these material containers rather than the raw sets of materials. Each type of material container brings a capability to the modeler. A primary design decision is to determine which material container is most appropriate for a set of materials at any specific point in the system being modeled. This decision is based both on the capability of each container to allow processing of its set of materials and on the capability of each container to provide for the movement of its set of materials to another place for additional processing.

The four primary types of material containers are **Parts, Bulk Containers, Streams,** and **Material Resources**. Each type of container is characterized by how it may move, how its contained set of materials may be accessed and initialized, and how it may be processed.

PWA-MATERIALS

The following are the attributes belonging to the class PWA-Materials which pertain specifically to waste management and which have been built into the model:

epa-hw - the EPA Hazardous Waste id number.

material-class - the hazard class of the material. The types of hazard classes are as follows:

TRU - Transuranic waste

LLW - Low level waste

TCW - Toxic chemical waste

NCW - Non-toxic chemical waste

MIXED - Mixed waste

SANITARY - Sanitary waste

RAW - Radioactive acqueous waste

recra-listed-material - a flag to indicate if the material is known to be RCRA listed when it appears in a waste stream.

units - the units of measure for the material. The model does not use the units of measure in any of its calculations. The units will appear on reports.

4.3 Parts

Parts represent discrete chunks of material that move from place to place and that only can be broken into smaller pieces by a processing step. There is no capability to move only some of the materials in a part.

Parts move from place to place under control of a **Workorder**. The Workorder specifies where the part will move, the processing step that will occur, and the time that the processing step will require. Workorders can also specify the duration of transfer of a part from one place in the facility to another place.

A user of PWA defines types of parts in a scenario. Individual instances of these types of parts will be created by the model when locations are created, when a processing step has a discrete part specified as an output, and by special events that allow the arrival of external material into the system.

When a user of PWA defines some class of part, the following attributes may also be defined:

footprint

The amount of storage space used by the part. See Capacity-Constrained-Locations for related information.

part-status

The part is **Stored** if the part is to reside in the initial location, and **Done** if the part is ready to be moved to the next location for processing.

workorder

The name of the class (or type) of the workorder which the parts should use for processing.

part-attributes

A user-supplied list of attributes which should be tagged to the part, i.e. color, manufacturer, or designer.

part-materials

An initial set of materials to be contained in the part. Note that if a part of this type is created as a result of a processing step, the processing step will determine the set of materials to be contained by the part. The initial part-materials attribute is used for parts that are created as the result of initializing a location, or by creation with an event.

Example:

```
:class MY-PARTS
    :parents (parts)

        :subclass TYPE-1-PART
            part-materials ((nickel 1) (copper 10) (iron 15))
            workorder cut-apart-workorder
            part-attributes ((color blue) (designer acme))
            part-status stored
            footprint 2
        :end-subclass TYPE-1-PART

    :subclass TYPE-2-PART
            part-materials ((copper 10))
            workorder cut-apart-workorder
    :end-subclass TYPE-2-PART

:subclass TYPE-3-PART
            workorder cut-apart-workorder
    :end-subclass TYPE-3-PART

:subclass TYPE-4-PART
    :end-subclass TYPE-4-PART
```

4.4 Bulk Materials

Bulk Materials represent a container of materials from which a subset (a batch) of the contained materials can be extracted for a processing step.

A bulk material container is created at a location and then remains at that location. It cannot move from place to place. A user must use some other transport mechanism to move the material in this container. For example, a user might specify a processing step that removes a batch of material from a bulk container and then places some of that material into a part. That part can then move according to its workorder. Also, the processing step can specify that material is to be directly added to a bulk container at the location of the processing step, or be added to a bulk container at some other location as a separate event. (See the :bulk and :feed output directives for workcenters.)

When the user of PWA specifies a bulk container, it is usually defined indirectly as a result of specifying the stored bulk containers of a location (see Locations). However, a user of PWA may wish to create special types of bulk containers in order to make use of triggering mechanisms for material amounts. Thus, the following attributes can be defined for a user-specified type of bulk container:

bulk-high-level-alarm

The material and the high quantity at which an alarm event will be posted.

bulk-low-level-alarm

The material and the low quantity at which an alarm event will be posted.

bulk-trigger-foreman

The name of a foreman object which will be notified via a trigger event when one of the high or low alarms is signaled.

Example:

```
:class MY-TANKS
    :parents (bulk-containers)

        :subclass saline-tanks
            bulk-high-level-alarm 250
            bulk-low-level-alarm 25
            bulk-trigger-foreman saline-supply-foreman
        :end-subclass saline-tanks
```

4.5 Streams

Streams are a pure transport mechanism (as one might think of a bulk container as a pure storage mechanism) that exist only to transport material from one processing step to a second processing step. There are no user-accessible attributes or definitions available for streams. The user can only specify a stream as the input or output of a processing step.

The major use of a stream is to model a true continuous flow (with no pipeline effects) where material flows through several processing steps in parallel. Processes producing and receiving streams operate in parallel with processing occurring simultaneously. All other material transport mechanisms in PWA occur sequentially.

4.6 Resources

Resources is a type of material container that represents material to be added to the model at a processing step. The conceptual model for these resources is a supply room or chemical warehouse where the location of the material is not important as long as any object in the simulation world can obtain the required materials. The main purpose of these materials is to provide a mechanism to add material to the simulation system in a controlled fashion.

It should be noted that material resources have several limitations. If these limitations become too restrictive, one should consider converting the form of the material container to a bulk container at some explicit location. In particular, material resources are considered to contain only one primary material type. (See trace elements in resources-with-amounts.)

4.6.1 Critieria for Identification of Resources

The user may define as many resources as desired within a PWA simulation. The following criteria may be used to determine which real-world resources should be used in the simulation model. A specific application of PWA may define additional criterion.

1. The amount used of the resource is of interest.
2. The resource is necessary for a operation of a workcenter.
3. Availability of the resource may constrain production.
4. The resource is a material that must be tracked in product, waste, and residue outputs.
5. The resource is a material that emits exposure to operators.

4.6.2 Unconstrained Resources

The simplest form of supply-room (resource-room) materials are those which do not constrain production. These resources act as if there is an unlimited supply of the material associated with this container. Unconstrained-resources are very simple as they can only contain one material which is the name of the resource. The user can only be interested in how much of this material was used.

The set of materials added to a processing step by a request to this resource is exactly the amount requested of a material of the same name as the class of resource. There can be no secondary materials or trace contaminates.

Attributes of resources that may be set by the user from a scenario are:

resource-amount-used

The total amount of the resource that has been used.

Example:

```
:class FACTORY-RESOURCES
    :parents (resources)
        :instance cobalt
        :instance water
            resource-amount-used 150
```

4.6.3 Constrained Resources

A subtype of resources is is a resource whose amounts are in limited supply. These resources are referred to as **resources-with-amounts**. Constrained resources are usually contained in a facility's stockroom or supply room which can hold a finite amount of any material. For example, the chemical stockroom can is large enough to hold 1500 liter bottles of hydrochloric acid. Thus, the amount of any one material held in the stockroom is in limited supply and thus can *constrain* production:

The following are examples of how constrained materials can be changed:

There may be a material returned other than a material with the same name as the instance of the resource being requested. e.g. The resource "dirty-oil" can now return a material other than "dirty-oil" such as "oil."

- The resource may now return a set of materials with more than one material. This allows trace contaminants to be in one's "oil."
- The resource may trigger a decision in the model when the amount of the resource goes below some level.

Attributes of resources-with-amounts that may be set by the user from a scenario are:

available-resource

The total amount of the resource that can be used.

resource-is-constraining

If the value of this attribute is set to **T** (the default,) the resource responds "NO" when asked if an amount of resource exists and there is insufficient resource to fill the request. A value of **nil** will cause the resource to always respond "YES" to such a request. A side effect of this is that the amount of the resource can go negative, and the amount of any extra-materials returned with the resource will be nonsense.

resource-material

The material to be provided as a result of this resource request. By default, the material returned is the same as the name of this instance of resource.

extra-materials

A set of materials that will act as trace elements to the primary material. When some fraction of the total amount of the primary material is provided by this resource, proportional amounts of each material in the extra-materials list will also be provided as trace elements. This attribute allows the user to initially "dirty up" a resource.

minimum-resource-trigger-level

When a particular material reaches a designated minimum level, the program is designed to trigger or alarm the user that the material (resource) has reached that minimum level so that appropriate actions can be taken. The amount below which a trigger event will be posted.

resource-trigger-foreman

The name of a foreman object upon whose behalf a trigger event will be posted when the amount of this resource drops below the minimum-resource-trigger-level.

Example:

```
:class FACTORY-RESOURCES-CONSTRAINED
    :parents (resources-with-amounts)
        :instance nickel
            available-resource 50

        :instance impure-acid
            available-resource 100
            resource-is-constraining T
            resource-material nitric-acid
            extra-materials ((iron 5) (nickel 2))
```

4.6.4 Materials

Materials may generate exposures to personnel who work with or around the materials. Any substance which can generate exposures can be defined as an instance of the class Materials.

exposures

A list of the types and amount of exposure generated by a material per hour of contact.

Example:

```
:class FACTORY-RESOURCES
    :parents (resources materials)
        :instance water
        :instance carbon-tetrachloride
            exposures ((carcinogen 1))
        :instance plutonium
            exposures ((penetrating-rad .05) (extremity-rad .01))
```

5.0 Reusable Resources

A number of types of resources exist which are used by a process which are not materials and which do not participate in the mass balance of a process. Such resources may be allocated and deallocated and must be available in sufficient quantity for a processing step to start. For example, an operator and a wrench may be needed in order to perform a particular process, but neither the operator nor the wrench will become part of the product outputs of the process. Such resources are termed reusable resources.

5.1 Critieria for Identification of Reusable Resources

The user may define as many reusable resources as desired within a PWA simulation. The following criteria may be used to determine which real-world objects should be in a simulation. A specific application of PWA may define additional criterion.

1. The amount of the resource used is of interest.
2. Availability of the resource may constrain production.
3. The resource accrues exposure that should be tracked and may affect resource availability.

5.2 Working Shifts

Shifts may be used to represent the times when a facility or part of a facility is working. Thus, the user may dictate that two hours are needed in a shift in order for a process to begin. In addition, operators may be associated with specific shifts, see OPERATORS.

The shift has a cycle of seven days, and within each day the user may define as many shifts as desired. Each day starts and ends at midnight. Shifts may begin the previous day, the first shift on Monday could start at 11:30 pm on the preceding Sunday night. Shifts may be contiguous, overlap or have gaps between them.

5.2.1 Basic Shift Attributes

The user may set the following attributes in a scenario:

holidays

A list of day month pairs when the shift is not available.

shift-schedule

A list of the start and end times for each shift by day. The syntax for shifts is:

Example:

```
:class factory-shifts
    :parents (working-shifts)
        :instance one-shift-per-day
            shift-schedule (
                (((0.0 -30 0) (0.0 30 7))) ;; Monday
                (((0.0 -30 0) (0.0 30 7))) ;; Tuesday
                (((0.0 -30 0) (0.0 30 7))) ;; Wednesday
                (((0.0 -30 0) (0.0 30 7))) ;; Thursday
                (((0.0 -30 0) (0.0 30 7))) ;; Friday
                () ;;Saturday
                ()) ;; Sunday
            holidays ((1 1) (4 7) (25 12))
```

5.3 Operators

Operators are groups of personnel of the same occupation or organization. Utilization is calculated on the pool. The user may define the pool to be one or more workers. Operators have some characteristics which are different from other resources in that they can be assigned to a shift, in varying numbers, and can accrue exposures.

5.3.1 Basic Operator Attributes

If the user does not define any attributes for a set of operators, the resource is treated as an unconstrained pool, and the hours worked by the operators are accumulated. By assigning other attributes, the operators will be treated as constrained resources.

The user may set the following attributes in a scenario:

total-operator-resource

The total number of operators in the pool. If there are any operators assigned to shifts, the total-operator-resource must be at least as great as the number of operators assigned.

accumulated-operator-time

The sum of the times the operators have been working. If desired this may be set in the scenario to represent initial conditions where the plant has an operating history which is being captured in the initialization.

shift-assignment

A list of the shifts and the number of operators assigned to each shift.

Examples:

Unconstrained Operator Pool

```
:class techs
    :parents (operators)
        :instance chem-techs
```

Constrained Operators

```
:class techs
    :parents (operators)
        :instance chem-techs
            total-operator-resource 5
```

Constrained Operators Assigned to Shifts

```
:class techs
    :parents (operators)
        :instance chem-techs
        total-operator-resource 5
        shift-assignment
            ((first-shift 3)
             (second-shift 2))
```

6.0 Exposures

Operators can accrue exposures via a number of different methods. If a pool of operators has received exposures such that the average exposure exceeds the exposure limits, then operators in that pool are no longer available.

6.1 Basic Exposure Attributes

accrued-operator-exposures

A list of the types and amounts of exposure which the operator pool has accrued. The accrual is for the entire pool.

limits-on-operator-exposures

A list of the types and limit for each type of exposure which the pool may accrue. The limit is per individual.

time-to-reset-exposures

A list of the type of exposure, the date on which each type should be reset to zero, and the periodicity with which it should be reset after the initial date.

6.2 Location-Time-Exposed-Operators

One of the methods by which operators can accrue exposure is by being in a location for a period of time. Exposures are accumulated based on the types of exposures generated in the location and the time spent in the location. This may be used to represent exposure to noise or background radiation. The types of exposure accrued will be based on the exposures specific to the locations in which the operators are assigned.

Example:

```
:class techs
    :parents (operators location-time-exposed-operators)
        :instance chem-techs
            total-operator-resource 5
            time-to-reset-exposures
                ((penetrating :yearly-calendar (0.0 0 0 1 1 1993)))
            limits-on-operator-exposures ((penetrating 5.0))
            accrued-operator-exposures ((penetrating 1.5) (noise 2.0))
```

6.3 Workcenter-Time-Exposed-Operators

One of the methods by which operators can accrue exposure is by being at a workcenter for a period of time. Exposures are accumulated based on the type of exposures generated at the workcenter and the time spent at the workcenter. This may be used, for example, to represent exposure to noise. The types of exposure accrued will be based on the exposures specific to the workcenter to which the operators are assigned.

Example

```
:class techs
    :parents (operators workcenter-time-exposed-operators)
        :instance chem-techs
            total-operator-resource 5
```

6.4 Contact-Exposed-Operators

One of the methods by which operators can accrue exposure is by being in contact with materials in a workcenter. Materials generate the exposure (see RESOURCES).

contact-exposures

A list of the types of exposure which the operator pool may accrue via contact with materials.

Example:

```
:class techs
    :parents (operators contact-exposed-operators)
        :instance chem-techs
            total-operator-resource 5
            contact-exposures (extremity penetrating)
```

6.5 Non-Contact-Exposed-Operators

One of the methods by which operators can accrue exposure is by being in the vicinity of materials in a location. Materials generate the exposure (see RESOURCES).

non-contact-exposures

A list of the types of exposure which the operator pool may accrue via being in the same location as materials.

Example:

```
:class techs
    :parents (operators non-contact-exposed-operators)
        :instance chem-techs
            total-operator-resource 5
            non-contact-exposures (extremity penetrating)
```

7.0 Locations

Central to all operations in the material-processing world simulated by PWA are the places between which sets of material move and which house the pieces of equipment that perform material transformations which represent processing. These places are called **Locations**.

Locations provide two primary services to PWA:

1. Locations have the ability to assess their current state whenever anything in their environment changes, i.e. when a piece of equipment finishes an operation, or material arrives or departs. Locations create events in the simulation model appropriate to the state changes.
2. Locations have the ability to constrain the movement of material into themselves. Regardless of whether the material moves as a part, as feed into a bulk container, or as material acquired and added by some processing step, a location constrains that material movement according to a wide range of user-selectable, and programmer-extensible movement constraints.

7.1 Critieria for Identification of Locations

The user may define as many locations as desired within a PWA simulation. An entire manufacturing plant could be modeled as one location. Every room in a plant could be modeled as a location. The user determines what locations should be modeled for the simulation to provide meaningful results.

The following criteria may be used to determine which real-world objects should be in a simulation. A specific application of PWA may define additional criteria.

1. The amount of material moving into and out of the location is of interest.
2. The amount of material stored within the location is of interest.
3. The amount of time required to move material from one place in the plant to another affects production.
4. The location has constraints on the types and amounts of material that may be in the location.
5. The location has constraints on the number and/or size of parts that may be in the location.
6. There is a relationship in the real-world between specific workcenters and bulk containers which should be maintained.

7.2 Unconstrained Locations

Simple locations, those without constraints, support the equipment that performs processing and the unconstrained movement of material in the facility.

7.2.1 Basic Location Attributes

The user may set the following attributes in a scenario:

stored-parts

A list of parts to be created at this location. The form is a list of specifications for new parts, each of the form (part-type number). The part-type is the name of a class of part, and the number is the number of parts to be created.

For example, the form:

```
:class locations
    :instance building-3
        stored-parts ((ingots 20) (molds 40))
```

creates a new location, named building-3, with twenty individual parts of type ingots, and forty individual parts of type molds. The part-materials of each individual part will be as described in the :class description of each part type elsewhere described in the scenario.

stored-bulk

A list of bulk-material containers at this location. The list can describe the containers in one of three ways:

1. (name)

   where name specifies the name of container which contains no materials.

1. (name (m1 a1) (m2 a2) ...)

   where name specifies the name of a container, followed by an initial set of materials. The name is used by pieces of equipment (workcenters) when they either require some set of materials from a bulk container in order to begin a processing step, or produce an output set of materials that will be placed into a bulk container.

2. ((type name) (m1 a1) (m2 a2) ...)

   where the name and initial materials are as in the second form, and the type is the name of a type of bulk container other than the default class bulk-containers. This form is used when a new type of bulk container has been created with a trigger-level and trigger-foreman.

Example:

```
:class locations
    :instance building-3
        stored-bulk
            ((liquid-gunk)
             (initial-feed (h2o 2000) (glue 150))
             ((monitored-output output-tank)))
```

This example creates three bulk containers at location building-3. The first will hold liquid-gunk, will contain no initial materials, and will be of type "bulk-containers." The second will be called initial-feed, will have some initial water and glue, and will also be of type "bulk-containers." The third will be called output-tank and will have no initial materials, and will be of a user-defined sub-type of bulk-containers called "monitored-output." The new type of bulk container "monitored-output" must be created by the user as a specialized type (see :subclass) that can have new attributes such as a trigger level set.

Note that the user does not create an instance of a bulk container. Just as with stored-parts, the object creation of specific bulk containers is performed automatically when the location is created.

attached-workcenters

A list of types of workcenters (pieces of equipment) at this location. The individual pieces of equipment will be created automatically when the location is created. Workcenters may be described in one of two ways:

1. ((type name) (type name))

   where type is the class of workcenter and name is a specific name associated with the workcenter.

2. (type type)

   where type is the class of workcenter. The location will generate names for each piece of equipment.

Example:

```
:class locations
    :instance building-3
        attached-workcenters
                ((turret-lathe lathe-a)
                 (turret-lathe lathe-b)
                 drill-press)
```

creates three pieces of equipment at location building-3. Two pieces of equipment are of type "turret-lathe" named lathe-a and lathe-b, and one piece of equipment is of type "drill-press." which will have a simulation generated name.These types must be described by the user in the scenario as subclasses of type "Workcenters."

exposures

A list of types of exposure and amounts. The amount is what one worker will receive by being in this location for one hour. The types of exposure are user specified names and should be related to the exposures which operators can accrue. See Operators.

Example:

```
:class locations
    :instance building-3
        exposures ((noise 5) (penetrating-rad .01))
```

7.2.2 Basic Location Capabilities

The user may directly cause events to occur within the simulation on behalf of locations. Events which may be directly posted are:

receive-parts

A command in the form

:at date-and-time (receive-parts part-type number-to-receive)

will cause the specified number of the class of parts to be directly placed into the location at the specified date and time.

Example:

```
:class warehouse
    :parents (locations)
        :instance receiving-dock
            :at (0.0 0 8 3 1 1992) (receive-parts ingots 25)
```

decide-operation

A command in the form

:at date-and-time decide-operation

will cause the location to assess its state and take appropriate actions. For example, if the simulation were initialized with parts stored in the location with appropriate workorders, the location would take appropriate action to move the parts where they should go. Similarly, if the location were initialized with stored-bulk, the location would take appropriate action to ensure the materials were processed, if possible.

Example

```
:class warehouse
    :parents (locations)
        :instance receiving-dock
            :at (0.0 0 8 3 1 1992) decide-operation
```

7.3 Constrained Locations

PWA provides several subtypes of locations that add additional functionality to the basic location. These new subtypes all add constraints on the movement of material from location to location.

A constraint on material movement provides a mechanism for a processing facility to develop bottlenecks. A constrained location will not allow material to move into that location if some constraint will be exceeded.

Note that the user need not say anything about the relationship between the class locations and the constrained-locations subclass. PWA already knows this relationship, and it would be an error for a user to attempt to change the relationship.

7.3.1 Unit-Capacity-Constrained-Locations

This subclass of location limits the number of parts that may be present in that location. This means that some parts and thus the contained sets of materials may not be able to move into some locations until space is available.

The user can set the following attributes in the scenario.

unit-capacity

The number of parts that can be present in a location at any time.

minimum-unit-trigger-level

A number of parts such that whenever a part leaves the location, causing remaining number of parts to be below the minimum, the unit-trigger-foreman will be activated.

maximum-unit-trigger-level

A number of parts such that whenever a part enters a location causing the number of parts in the location to exceed the maximum, the unit-trigger-foreman will be activated.

unit-trigger-foreman

The name of a foreman object to be notified as described above.

Example:

```
:class unit-capacity-constrained-locations
    :instance building-3
            unit-capacity 10
            minimum-unit-trigger-level 2
            maximum-unit-trigger-level 9
            unit-trigger-foreman building-3-warning-foreman
```

7.3.2 Capacity-Constrained-Locations

This type of location limits the amount of floor space available for parts. Each part has a footprint attribute that represents the amount of floor space occupied by that part. The location will not allow a part to move in unless there is floor space available for part.

The user can set the following attributes in the scenario:

storage-capacity

The amount of floor space available.

minimum-capacity-trigger-level

The amount of floor space utilized such that whenever a part leaves a location and the occupied floor space falls below the minimum, the unit-trigger-foreman will be activated.

maximum-capacity-trigger-level

The amount of floor space utilized such that whenever a part enters a location and causes the occupied floor space is exceed the maximum, the unit-trigger-foreman will be activated.

capacity-trigger-foreman

The name of a foreman object to be notified as described above.

Example:

```
:class capacity-constrained-locations
    :instance building-3
                storage-capacity 10
                minimum-capacity-trigger-level 2
                maximum-capacity-trigger-level 9
                capacity-trigger-foreman building-3-warning-foreman
```

7.3.3 Material-Constrained-Locations

This type of location limits the amount of various combinations of materials which may be in the location. This constraint applies to bulk containers, resources acquired during a processing step, and the material in parts.

The user can set the following attribute in the scenario:

critical-materials

A list of materials and associated capacities. A single material and a corresponding capacity will limit the movement of that material. A set of material names and a capacity will limit the sum of those materials.

Example:

```
:class nasty-places
    :parent material-constrained-locations
        critical-materials (
                ((Pu U) 500)
                (acetone 50))
```

Any instances of nasty-locations can have no more than a combined total of 500 units of the materials Pu and U, and no more than 50 units of acetone. The first might model a radioactive limit, and the second might model a toxic chemical limit. PWA contains no inherent knowledge about either radioactivity or chemical exposure, but the analyst could use this feature to model such situations.

The constraints are independent. In the example above, a part might be kept out because it contains uranium and exceeds the (Pu U) constraint while another part might be let in because it contains acetone, and the acetone constraint has not yet been exceeded.

7.3.4 Maximal-Footprint-Locations

This type of location limits the size of the largest single part which can be in the location. See Parts for the part footprint.

The user can set the following attribute in the scenario:

maximal-footprint

The size of the largest part which may be in the location. The number must be in consistent units with the footprint attribute of the part.

Example:

```
:class small-places
    :parent maximal-footprint-locations
        maximal-footprint 2.5
```

7.3.5 Mixing Types

Additional types of locations may be created by specifying more than one type of locations as a parent of the class.

Example:

```
:class warehouses
    :parent unit-capacity-constrained-locations
        unit-capacity 450

        :instance building-20

        :instance building-21

        :instance building-22
            unit-capacity 275


:class factories
    :parents (unit-capacity-constrained-locations capacity-constrained-locations)

        :instance assembly-building
            unit-capacity 100
            storage-capacity 200
```

Two types (classes) of locations will be created.

The first class is named warehouses and is a unit-capacity-constrained-location. Buildings 20 and 21 will inherit the unit-capacity of 450 from the warehouse, while building-22 resets the capacity to 275.

The second class named is the class of all factories. This type of object is both a unit-capacity-constrained-location and a capacity-constrained-location. The instance assembly-building inherits the capabilities and the attributes from both parents of factories.

8.0 Workcenters

After a material, in any of its containers, has moved into a location, a piece of equipment located at that location may process the material. Defining a process for a set of materials involves specifying a set of materials as input, specifying some set of material containers to receive the output materials, and specifying the portions of the input materials that will be placed into each of the outputs. A piece of equipment transforms sets of input materials into sets of output materials in output containers while maintaining a mass-balance.

Objects that represent these pieces of equipment are instances of the class **workcenters**. The class that will be used for all equipment in PWA/TASP is a direct subclass of workcenters called **mass-balance-workcenters.**

The user of PWA does not directly create instances of workcenters in scenario files. Rather, the user describes the equipment desired and then names these classes in the attached-workcenters attribute of a location. When the location is created, it creates the appropriate pieces of equipment that perform the desired processing.

8.1 Critieria for Identification of Workcenters

The user may define as many workcenters as desired within a PWA simulation. A workcenter may represent one piece of equipment, one workstation, or a set of equipment and workstations. The user determines how workcenters should be modeled for the simulation to provide meaningful results.

The following criteria may be used to determine which real-world objects should be in a simulation. A specific application of PWA may define additional criteria.

1. The types and amounts of material output by the workcenter are of interest.
2. The types and amounts of resources used by the workcenter are of interest.
3. The time worked by the workcenter is of interest.
4. The workcenter is a critical piece of equipment that may constrain production.

8.2 Basic Workcenter Attributes

Several attributes of workcenters may be described by the user:

workcenter-processing-time

The default amount of time that this piece of equipment will require to perform a processing step when it cannot be determined by other factors such as a part's workorder.

This is typically used when the only input to the processing step is from the chemical stockroom.

exposures

Types and amounts of exposures generated.

breakdown-random-deviate
repair-random-deviate

Random-number generating objects used to drive stochastic breakdown and repair times of this piece of equipment.

failure-random-deviate

The name of a random-number-generating object used to drive the number of operations performed between failure to produce a correct output.

process-step-descriptions

A list of the different behaviors of this piece of equipment.

While most of the other attributes of objects in PWA are simple in structure, the process-step-descriptions attribute of a workcenter is a very complex data structure.

Example:

```
:class mass-balance-workcenters
    :subclass lathe
        workcenter-processing-time (0.0 0 2)
        exposures ((penetrating-rad .05))
        breakdown-random-deviate lathe-normal-breakdown
        repair-random-deviate lathe-normal-repair
        failure-random-deviate lathe-normal-failure
        process-step-descriptions ()
    :end-subclass lathe
```

8.3 The Process-Step-Descriptions Attribute

Any single piece of equipment in PWA is able to perform only one sort of processing at any one time. However, a piece of equipment may be able to perform several different types of processing at different times depending on available inputs.

For example, a drill-press may be able to process any item that needs to be drilled. Drilling a simple piece of wood may require nothing more than the input piece of wood and an operator. Drilling some exotic alloy might require a special drill bit (of which only a few might exist), special cutting oil from the stock room (which may go into some waste-processing facility for re-processing), two certified, specially-trained operators and an EPA auditor (who only works on Tuesday and Thursday afternoons).

These independent descriptions of processing steps are gathered onto a list that is the value of the attribute process-step-descriptions of the mass-balance-workcenter. Thus, the form is:

```
:class mass-balance-workcenters
    :subclass drill-presses
        process-step-descriptions (
            <description 1>
            <description 2>
            ...)
    :end-subclass drill-presses
```

When the state of a location changes, a part of the location's assessment is to allow each workcenter at that location to see if it is able to begin a processing step. The mass-balance-workcenter does this by examining each of the descriptions on its process-step-descriptions list in order to find one whose starting and ending conditions are such that it can process.

There might be many reasons why a process step might not start. Examples include:

1. Some required input part might not be available.
2. There may be insufficient material in a bulk container to provide a minimal input batch.
3. Some material resource from the stockroom may not be available.
4. An output that transports material to another location might not be allowed because the transport of the material would violate a constraint at the receiving location.

From this point on in this section, the description of a single process step from the list of all process-step-descriptions will be emphasized.

8.4 Descriptions of Process Steps

Each description is a list of six, seven, or eight elements. These elements must be presented in the following order:

1. Initial status

   A state in which the workcenter must be in order for this description to be considered.

2. Input parts, stream, or bulk container

   a. A list of parts and associated tags or labels such that the part can be referenced later in the description if needed, or

   b. A list of bulk containers and descriptions of the batches of material to be taken from each, or

   c. A list describing a single continuous stream.

3. Currently ignored

   Must be "()" or "nil".

4. Material resources

   A list of materials to be added from the stockroom.

5. Reusable resources

   A list of resources which are acquired and then returned at the end of the processing step. Their materials do not participate in the material balance between inputs and outputs for the process step.

6. Normal output
7. Failure output [optional]
8. Breakdown output [optional]

   Each of the three outputs is of identical form, and will be described below simply as outputs. Each contains a final state in which to leave the workcenter at the end of the processing step, and a set of outputs and the associated mass balance of all the materials in any of the inputs.

Each of these sections will now be described in more detail.

8.5 Initial Status

The workcenter is created in state **Free**. Each output will indicate the state in which to leave the workcenter. Often this is free. However, the user can place the workcenter in some different state. For example, a furnace might start in state free, and then:

1. perform a step that leaves it in state heating,
2. perform a heating processing step that leaves it in state melting,
3. perform a melting step, that leaves it in state cooling,
4. perform a cooling step,
5. at the end of the cooling step, leave the furnace once again in state free.

The user must describe each of these processing steps as a description in the list of process-step-descriptions. Each description will have a different starting state representing the progression of the furnace through its activity.

8.6 Inputs

The set of inputs for a description specifies those material containers that must exist at the location for processing to be performed according to this description.

The inputs can be a set of parts, a set of bulk containers, or a single continuous stream. These three forms cannot be mixed.

The sets of materials contained in these inputs will be merged into a single set of materials by the workcenter and then distributed to the outputs according to the material balances as specified later in the section.

There may be no input parts, bulk containers, or streams listed as inputs. The form:

()

may be used for the input specification. The set of materials for the process must come from the material resources (described below) and the processing time will be the Workcenter-Processing-Time attribute of the workcenter.

8.6.1 Parts

A mass-balance-workcenter can process a set of materials from one or more parts. The parts must be located at the location containing the workcenter and must have a part status of **stored**.

The parts are specified with two elements. The specification of one of the part inputs to a description is

(part-label processing-type)

The **part-label** is a name for this part to be used whenever this description needs to refer to the part. The part-label may be used to identify the part when the part is to continue on as an output of this processing step, and when the workorder and attributes of the part are copied to a new part created for output from this processing step.

The **processing-type** names a processing step that must match the processing step named on the part's workorder. The purpose of this type of specification (rather than e.g. just naming the type of part desired) is to allow the piece of equipment represented by the workcenter to process different types of parts. The processing-type * is a special case and indicates that the process step description applies to any part that in the location.

For example, a drill press that can drill both 2x4s and exotic-metal-ingots can be described as a workcenter that can take any part whose workorder says "drill-hole." The input specification for this drill press might look like

(part1 drill-hole)

The input will be a part called "part1" for the remainder of this process-step description and that the part to be chosen must have a step "drill-hole" as the current entry in its workorder. Either a 2x4 or an exotic-metal-ingot can have a workorder that says "go to this location and drill-hole" and either can be drilled by this workcenter.

If this were the only part required for this processing step, the full input specification would be

((part1 drill-hole))

Note the extra parenthesis. The first pair says "This is the set of all input specifications." The second, inner, pair says "This is the single specification for drilling part1."

Several parts can be specified, each with a different workorder step. Thus, a complete specification for something that requires several different parts (such as constructing a box where top and bottom are different and all sides the same) might look like the following:

```
(   (top assemble-top)
    (bottom assemble-bottom)
    (side1 assemble-side)
    (side2 assemble-side)
    (side3 assemble-side)
    (side4 assemble-side)
)
```

8.6.2 Bulk Containers

A mass-balance-workcenter can process a set of materials from one or more bulk containers located at the same place as the workcenter. The major difference between the use of a part and the use of a bulk container for input is that the set of materials in a part is processed as a whole, with no ability to break that set of materials into subsets, while the processing of materials from a bulk container is performed by determining a batch of material that is a sub-set of the total materials in the container.

The form of the bulk container input specification is

(:bulk container-name basis amount-descriptor amount time min max)

where

container-name

the name of some bulk container at this location.

basis

either the name of some material in the bulk container or the symbol TOTAL-MASS. The basis for the batch is used to allow all materials in the container to be used for input in an amount proportional to the basis for the batch. This means that is a batch of material based on some material (e.g. Iron) is chosen such that the batch uses 10% of that material in the container, then all other materials in the set of materials in the bulk container will participate by allocating 10% of their amounts to the processing step.

amount-descriptor

describes size of the batch. The amount descriptor must be one of the forms:

amount

When amount is just a number, this will be the batch size. The fraction of the material in the bulk container that will be used in this processing step is computed by comparing this amount of the basis material (or the sum of all the materials for TOTAL-MASS) with the total amount of the basis material in the container. Thus, if the bulk container contains (Fe 1000) in its set of materials, the basis is Fe, and the amount is 100, then 10% of each of the materials in the container will be used for this batch.

(material fraction)

material names some material in the bulk container, and fraction is a number between 0.0 and 1.0. This fraction of the material named is taken. The material typically is the same as the basis material.

(TOTAL-MASS fraction)

This acts just like the previous form except that the amounts of all materials in the bulk container are added together to get the amount that will be multiplied by the fraction.

amount and time

Provide a rate at which the batch will be processed, and allows the determination of the processing time for this batch. As a shorthand and to avoid the roundoff errors associated with this computation, if the amount is "()" then the time is taken as the absolute time for the batch regardless of the amount.

min

The minimum batch size that will be processed. It is optional and, if omitted, will be zero indicating that any amount of material remaining in the container will be processed regardless of how little there is.

max

The maximum amount that will be processed as a batch. This is optional, and if omitted will be the normal batch size.

An example is:

(:bulk tank TOTAL-MASS (TOTAL-MASS 0.5) 10 (0.0 30) 0 100)

which says "We will always try to do half the mass in the tank, but when the tank is very full, we will never do more than 100 units in a batch.

8.6.3 Streams

The materials for processing can come from a continuous stream created as the output of some other processing step. There can only be one of these specified at a time. The form is

```
(:stream name)
```

where the name is specified when the stream is created as an output from some other workcenter.

8.7 Material Resources

Material resources are described after the other inputs so that the resource amounts allocated can be computed from the materials in the other inputs. This reflects the notion that the material resources represent some sort of chemical stock room that provides materials in support of the primary inputs to the processing step.

All resource requests will reference a resource object. This is an instance of some subclass of resources defined in the scenario. The user will always phrase a request to the resource in terms of some absolute amount of that resource. The resource will provide a set of materials to the process based upon that absolute amount and the amount of the resource already existing in the input to the processing step. Resources can be defined that provide behavior ranging from very simple (give back one material with the same name as this resource in the amount requested) to the complex (give back some set of materials that does not include a material with the same name as the resource.)

The material-resources portion of the process-step description is a list of resource requests. All of the resource requests must be satisfied in order for the process to proceed. This means that the resources requested must all have sufficient amounts available to fill the requests.

A request for material resources can take one of two forms:

(resource-name amount-specification)
(resource-name :absolute amount-specification)

where resource-name is the name of some instance of a resource defined in the scenario, and amount is in one of four forms.

1. An absolute amount. The form of this is just a number. This represents a request like "Get 2 liters of acid from the stock room!"
2. An amount relative to some material already in the input due to parts, bulk, or a stream. The form is:

   (material multiplier)

   The amount of the named material that is in the input from the input parts, bulk, or stream, is multiplied by the multiplier to get the amount to be requested. This represents a request like "Get 2 liters of acid for each metric ton of iron!"
3. An amount relative to the total mass of all materials in the inputs. (Note the assumption that all material amounts in the input material set are consistent.) The form is:

   (TOTAL-MASS multiplier)

The amounts of all materials in the input is added together and then multiplied by the multiplier to get the amount to be requested. This represents a request like "Get 2 liters of acid for each 100 grams of total material!"

4. An amount relative to the total time that will be taken to process all materials in the inputs. The form is:

   (TOTAL-TIME multiplier time)

   The time that will be taken to process the input is multiplied by the multiplier and divided by the time-interval to get the amount to be requested. The multiplier and the time taken together provide a rate. This represents a request like "Get enough acid to support 2 liters per hour while we are processing!"

Each of the four forms turns into a request for some resource (the resource named at the start of each specification) and an amount of that resource. The behavior of the resource is to return enough of the resource's primary material so that the total amount of that material in the inputs is the amount specified by the resource request.

For example, consider a request made to a resource named *clean-oil*, whose resource-material attribute is set to the material name *oil*, in the amount of 10 units. A request for this resource could be (clean-oil 10). If the set of materials provided by the parts, the bulk containers, or the stream providing the materials as input to this processing step already have 5 units of the material oil, then only 5 more units would be added by the resource. If there are more than 10 units of oil already in the input materials, then no oil will added.

A request for a resource (resource amount) should be read as "Give me enough of the resource's primary material and any other trace elements contaminating it such that total amount of the primary material in the inputs is greater then or equal to the requested amount."

The **:absolute** indicator in the second form of the resource request indicates that the amount requested is to be provided regardless of the amount of that material already existing in the input to the process.

Simple resources can be combined to produce the full set of resources required. At the top level of the request for material resources, this is an implicit conjunction of the resource requests. Thus, the form of the material-resources portion of the process-step description is:

```
(
    <request-1>
    <request-2>
    ...
)
```

Each resource request is either a simple request as described above, or a set of simple requests combined with AND and OR. For example, if one wishes for resource-1 OR resource-2 OR resource-3, the following form is used:

```
(OR <resource-1>
    <resource-2>
    <resource-3>)
```

Within these forms, additional combinations using AND and OR can be used.

```
(
   (sand 10)
   (OR (AND     (dirty-oil 5)
                (clean-oil 5))
       (clean-oil 10))
)
```

requests two resources. The first is for ten units of sand. In addition to the sand either five units of dirty oil and five units of clean oil are needed. However, if those two are not available, ten units of clean oil can be used.

8.8 Reusable Resources

Reusable resource have the same form as material resources in the description, and specify requests to instances of the resource type. However, reusable resources are not incorporated into the materials being processed by the workcenter as are material resources.

Rather than take the materials provided by the resource and incorporate them into the set of materials that will be distributed to the outputs, at the end of the processing step the resource and its associated materials are returned to the resource unchanged. Consequently, the emphasis is on the resource itself rather than the materials provided by the resource.

For example, a resource request

(wrench 2)

where the wrench is an instance of the class resources-with-amounts will result in two of the available wrenches being used by this piece of equipment for the duration of the processing step. The two wrenches never participate in the material balance performed by the workcenter. The two wrenches are returned to the resource at the end of the processing step.

Reusable resources are used in the model to constrain equipment operation where the items that constrain the production are not distributed to the outputs of the processing step.

Examples are

- tools

  such as the example of the wrenches.

- operator pools

  where available personnel are shared by many pieces of equipment.

- shift time

  where for example, in order to start a process, 52 hours must be left before everyone goes home Friday night. The notion of getting 52 hours from the stock room and then giving it back when finished is meaningless, but serves to constrain the beginning of a processing step.

8.9 Normal, Failure, and Breakdown Outputs

Each description has one, two, or three output specifications.Each output has an identical in form (to be described below) and are used to provide different process outputs depending on the selected ending condition of the equipment.

The first output (that must always exist) is considered the "normal" output behavior of the description. This will be the default behavior unless one of the other two is chosen.

The second output is the output behavior to be used when the processing step has "failed" for some reason. This behavior is driven stochastically by the failure-random-deviate attribute of the workcenter, and is expressed as number-of-operations until a failure. Note that a failure need not have the negative connotations of a true ruined processing step. This feature can be used when a processing step produces some fraction of its output as output type 1 and the rest output type 2.

The last output is the output behavior to be used when the piece of equipment breaks down. In order for this output to be used, there must also exist a value of the breakdown-random-deviate attribute of the workcenter that gives a stochastic object (from class random-deviates or one of its subtypes) that will provide a time of the next breakdown. When this time is reached, the breakdown output will be chosen instead of the normal output, and a repair will be scheduled according to the repair-random-deviate attribute of the workcenter.

8.10 Outputs

Each output (normal,failure, and breakdown) has the same form. Each is a list of three or more elements.

1. The state in which to leave the workcenter at the end of the processing step.
2. A material name to be used as the default output material in order to drive the default material default distribution rule described below. This can be "()" to disable this rule.
3. (And 4. and 5. and ...) Individual output specifications that describe the portion of the input materials that go to each particular output, and the type of material container or material transport mechanism to be used to contain these materials.

The third (and fourth, and fifth, and ...) sections describe the disposition of materials. Several forms can be used here in order to describe different material containers or material transport mechanisms for housing the output materials.

The fundamental characteristic of the output specification is that all materials in the input (from parts, bulk containers, a stream, and material resources) must be accounted for in the output. The material balance (sometimes called mass balance although the material amounts in the set of materials being processed need not be masses) can be specified in full for each material, or one of two defaulting rules can be used.

The different forms of output specification are

1. (Part-Label Material-Balance)
2. (:new Part-Type Workorder Attributes Material-Balance
3. (:stream Location Type Material-Balance)
4. (:bulk Container-Name Material-Balance)
5. (:feed Location Container-Name Material-Balance)
6. (:resource (Resource-Specification) Material-Balance)

Each output specification describes one type of material container or material transport mechanism, followed by a material balance specification. The material-balance specification is of identical form for each output specification type, and will be described later. At this point is sufficient to note that the material balance will specify what fraction or amount of each material in the set of input materials will be placed in this material container or on this material transport mechanism at the end of this processing step.

8.10.1 Existing Part Output

One possible output is to take an input part and pass it on from this processing step. The set of materials that was contained in the part at the start of the processing step will be replaced with the set of materials described by the Material-Balance associated with this output specification. The form of this output is:

(Part-Label Material-Balance)

The existing workorder and part attributes of the part remain. The part will proceed to its next processing step as specified by its workorder.

Note that the part-label used to name this part is the label associated with the part when it was specified as an input. This use of the part-label is one of its primary uses.

Any input parts that are not passed on in this way are destroyed, and disappear from the model. The sets of materials contained in those parts remain to be distributed at the end of the processing step. Only the container vanishes.

8.10.2 New Part Creation

A new part may be created to contain the set of materials specified by a material balance. The form of this output is:

(:new Part-Type Workorder Attributes Material-Balance)

The Part-Type is the name of some class of part. An instance of this type of part will be created to contain the set of materials specified by the Material-Balance. The part-materials attribute of the part-class will be overridden by the set of materials produced by this processing step.

As this is a new part, a workorder and part-attributes list must be specified.

The Workorder is the name of a class of workorders. An instance of this workorder will be created and added to the new part. The workorder may be the part-label of an input part in order to state that this new part should get a copy of the input part's workorder. This is useful when (e.g.) an existing part is cut into two half parts represented by a different part class. Each half part might get a copy of the original part's workorder which would describe the process followed by, not only the original part, but also by its two halves.

The Part-Attributes is a list of attributes to be added to this part. The Part-Attributes may be "()" to specify no attributes. The part-attributes can be the part-label of an input part in order to state that the new part should get a copy of the input part's attributes. Finally, a new list of attribute value pairs can be created by listing the new attributes and values. A part-label from an input part can be used in two places in this new list. It can be at the same top level as the new attribute value pairs and include all the attributes of the labeled part. A part-label can also be used as a value for an attribute value pair. This use creates a reference to the original part as the value of the attribute.

Some examples of part-attribute specifications are:

- part-1

  Where part-1 is the label of one of the parts input to this process step. All the attributes of that part will be copied to this new part.

- ((color blue red) (builder acme-factory-7))

  The new part will have a new attribute list with to properties, color and builder. The value of color is (blue red) and the value of builder is acme-factory-7.

- (part-1 (color blue) part-2)

  Where part-1 and part-2 are part-labels of two input parts to this process step. The new part will get a new set of attributes consisting of all the attributes from part-1, followed by a color attribute with a value of blue, followed by all the attributes of part-2. As attributes are always searched in order, this will effectively override a color attribute from part-2.

- (part-1 (color blue) (original-parts part-1 part-2))

  Where part-1 and part-2 are part-labels of two input parts to this process step. The new part will get a new set of attributes consisting of all the attributes from part-1, followed by a color attribute with a value of blue, followed by an original-parts attribute whose value is the list of values (<part-labeled-by-part-1> <part-labeled-by-part-2>).

8.10.3 Continuous Stream Creation

The set of materials described by the material balance may be transported to some location to be processed in parallel with this processing step. The form of this output is:

(:stream Location Type Material-Balance)

The Location is the name of a location where some piece of equipment must exist with a process-step description specifying this stream as an input.

Before processing is started, the workcenter determines that the receiving workcenter exists and can process this output and that the receiving location can accept the material according to its constraints. This determination is part of the starting criteria for this processing step. Once this processing step is started, the receiver begins processing the material at exactly the same time. The processing of the material proceeds in lockstep.

Chains of :Stream outputs can be constructed to simulate a series of pieces of equipment that process a flow of material in a continuous fashion.

8.10.4 Bulk Container Output

There are two mechanisms that place output in a bulk container. They differ in that one specifies a bulk container in the same location as the workcenter as an output, and the other specifies a transport step to (possibly) some other location with the material being placed in a bulk container at the destination location. The operational difference that the analyst will use to determine which form to use is that the more complex form uses an additional event in the simulation to perform the transport, and this event is audited by, and reported upon, by the reporting mechanism of the model. Examining this transport event and its associated reports allows an analyst to examine those material flows associated with this type of output.

The simple form is

(:bulk Container-Name Material-Balance)

This specifies that the set of materials described by the material-balance will be placed in a bulk container named Container-Name at the location where the processing step is occurring. No explicit transport takes place.

The form that includes a transport step is

(:feed Location Container-Name Material-Balance)

This form will post a Receive-Feed event for Location that will transport the set of materials specified by the Material-Balance and place them in the bulk container Container-Name at that location. This transport will occur even if the destination is the same location as that where the processing step is happening.

8.10.5 Resupply of the Stock Room

A set of materials can be returned to the stock room in order to resupply some material resource. The form of this output is:

(:resource (Resource-Specification) Material-Balance)

The material balance describes the set of materials to be returned. The materials in the set described by the associated material-balance will be given to the resource, and if the resource supports trace materials, they will serve to contaminate the "pure" resource.

The Resource-Specification can be one of four forms:

1. (name amount)

   Resupply the material resource "name" with the amount specified.

2. (name material multiplier)

   Resupply the resource "name" by an amount computed from some material associated with this output by the material balance. The amount of the "material" in the output will be multiplied by "multiplier" to produce the desired resupply amount.

3. (name TOTAL-MASS multiplier)

   Resupply the resource "name" by an amount computed from all materials associated with this output by the material balance. The amounts of all materials in the output will be added (with an assumption that units are compatible!) and the resulting sum will be multiplied by "multiplier" to produce the desired resupply amount.

4. (name TOTAL-TIME multiplier)

   Resupply the resource "name" by an amount computed from the amount of time taken by this processing step. The amount of time taken by this processing step will be multiplied by "multiplier" to produce the desired resupply amount.

8.10.6 Material Balances

Each output receives a share of the input materials for the processing step. The basic condition maintained by the mass-balance-workcenter is that all materials input to a processing step are accounted for in the output of that processing step. It is the material-balance specification for each output that allows the user to control the portions of each material that appear in each output.

The material balance is a list of material names (from the set of materials that is being processed by the workcenter) and a specification of either the absolute amount, or a percentage of the total of that material that is to appear in the output associated with the material balance specification.

There are two separate types of defaulting rules for allocating any materials that are not explicitly mentioned in any of the output's material balance specifications.

There is the ability to transform one set of materials into another material in order to change the level of detail being examined by the model or to simulate chemical reactions amongst the input materials.

Apportioning materials to outputs is performed in three steps.

1. Material is allocated to outputs based on specifications of absolute amounts in any of the material balances.
2. Material is allocated to outputs based on specifications of material fractions, in any of the material balances.
3. Two defaulting rules are used to apportion any materials not mentioned in the material balances.

8.10.7 Absolute Amounts in a Material Balance

The first distribution of material to the outputs according to the material balances is performed by examining each material-balance entry for an entry of the form:

```
(material AMOUNT number)
```

Where "material" is the name of a material in the input, and "number" is the absolute amount of the material to be placed in the output.

8.10.8 Material Fractions in a Material Balance

The second distribution of material to the outputs according to the material balances is performed by examining each material-balance entry for an entry of the form:

```
(material number)
```

Where "material" is the name of a material in the input, and "number" is the fraction of the material to be placed in the output. The number must be between 0.0 and 1.0.

8.10.9 Defaulting Rules

After material is distributed amongst the outputs according to absolute amounts and fractions, the input set of materials is examined to ensure that each material distributed according to either of the first two rules has been completely distributed so that 100% of the material is accounted.

There may be materials remaining which were not specified in the material balance. The output description is examined to determine which of three actions should be taken. The three actions are:

1. The Default Material Rule,
2. The Default Output Item Rule,
3. An Error stating that the materials cannot be balanced.

8.10.10 The Default Material Rule

If the chosen output specification of the process-step description has a default output material name as the second element, then that material name is considered to be the "primary" material of this processing step, and all unmentioned materials in the output will be distributed to the outputs in the same proportion as that primary material.

For example, consider a processing step where the input materials are:

```
(Fe 100) (Ni 10) (C 6)
```

And the chosen output description is:

```
(   Free
    Fe
    (:new half-ingot wo () (Fe 0.5))
    (:new half-ingot wo () (Fe 0.5)))
```

The processing step will produce two new parts of type half-ingot, each with 50% of the iron from the input. The Nickel and Carbon will follow the Iron proportionally. Thus, each half-ingot will contain the following material set.

```
(Fe 50) (Ni 5) (C 3)
```

8.10.11 The Default Output Item Rule

It is possible to specify one (and only one) output description as a default output for any unmentioned material. Placing an indicator ":default" before the material balance of an output specifies that this output is to receive any materials not mentioned in the material balances of the outputs. The Primary Material for that output description must be "()" so that there is no confusion over which defaulting rule to use.

For example, given the example above, an output description of:

```
(
    Free
    Fe
    (:new half-ingot wo () (Fe 0.49))
    (:new half-ingot wo () (Fe 0.49))
    (:bulk waste :default (Fe 0.02))
)
```

will place 49% of the iron into the first part, 49% into the second part, 2% into a waste bulk container, and then put all of the Nickel and Carbon into the waste. This might represent an electro-refining step that removes impurities from the input.

8.10.12 Material Transformations

As a simulation model of a large processing facility runs, the facility will be examined at different levels of detail depending on the analysis being performed. For example, in one portion of a processing facility, it might be desired to track elemental materials such as Iron, Carbon, and Nickel. In some other portion, it might be desired to track some combination such as Steel. This level-of-detail change can be accomplished within the material balance.

Rather than a simple material name as the first item in a material balance specification, a first element that is the list

```
(new-material amount-specification)
```

indicates that this output specification describes a transformation of one set of materials into a new material. The full item in the material balance is then

```
((new-material amount-specification) material-balance)
```

The "material-name" will be the name of the new material. (The term new material here should not imply that the material must exist nowhere else. Rather, it is quite reasonable to make (e.g.) Rust out of Iron and Oxygen while additional Rust already participates in the processing step.)

The amount-specification describes how much of the new material to make relative to the set of materials in the inputs. The forms of the amount-specification are:

amount

(material multiplier)

(TOTAL-MASS multiplier)

(TOTAL-TIME multiplier)

Where the material reference will refer to the amount of that material in the material-balance for the transformation specification only.

The material-balance portion of the transformation specification is a complete and independent material balance that participates with the output of the workcenter with the same status as every other material-balance for this processing step's output. In particular, the material balance may start with a :default so that all unmentioned materials will participate in the material transformation. This might be useful when one wishes to make material disappear. For example:

```
(
    Free
    ()
    (:new half-ingot wo () (Fe 0.49) (Ni 0.25))
    (:new half-ingot wo () (Fe 0.49) Ni 0.25))
    (:bulk waste (Fe 0.02) ((trash (total-mass 1.0)) :default (Ni 0.5)))
)
```

This indicates that the waste container gets 2% of the iron, and a new material called trash whose amount will be the sum of all materials incorporated and which will collect everything in the input other than the Iron and the Nickel that goes to the half-ingots. The bulk container will contain only Fe and Trash. The mention of Nickel in the material balance for trash is due to the mention of Nickel in the two half ingots. The default output of trash will only affect those materials (such as Carbon) that are mentioned nowhere in any material balance. Since Nickel was mentioned somewhere, it must be fully balanced.

As the material balance associated with a transformation specification is a complete material balance, it can, itself, contain transformations. While such transformations within transformations are legal, it probably indicates a process that should be split into more manageable steps.

8.11 Process-Step-Descriptions Examples

```
:class test-factory
    :parents (mass-balance-workcenters)

    :subclass inspect-part-wc
        process-step-descriptions
            ((FREE
                ((part-1 inspect))
                ()
                ((water 1.5))
                ((inspectors 1))
            (FREE
            iron
            (part-1)
            ))
    :end-subclass inspect-part-wc


:subclass weld-wc
    process-step-descriptions
    (
        (FREE
            ((part-1 weld-right)
            (part-2 weld-left))
            ()
            ((solder .25))
            ((certified-welders 1))
            (FREE
                iron
                (:new welded-part part-1 ((toxic-metals yes) part-1) (iron 1.0))
        )
        (FREE
            ((part-1 weld-iron-rod))
            ()
            ((solder .25)
            (iron .5))
            ((certified-welders 1))
            (FREE
                iron
                (:new welded-part part-1 ((toxic-metals no) part-1) (iron 1.0))
        )
    )
    :end-subclass weld-wc
```

```
:subclass cut-apart-wc
   process-step-descriptions
        ((FREE
              ((part-1 cut))
             ()
             ((or (clean-oil .5)
                   (recycled-oil .75))
             ((certified-machinists 1))
        (FREE
        iron
        (:new half-part part-1 ((original-part part-1) part-1) (iron .49))
        (:new half-part part-1 ((original-part part-1) part-1) (iron .49))
        (:bulk scrap-turnings (iron .02))
        ))
:end-subclass cut-apart-wc


:subclass cast-wc
   process-step-descriptions
        ((FREE
              ((:bulk molten-metal-vat iron 100 nil (0.0 10) 50 150))
             ()
             ()
             ((foundry-worker 1))
        (FREE
        iron
        (:new casting store-ingot-wo () (iron .98))
        (:feed foundry remelt-scrap (iron .02))
        ))
:end-subclass cast-wc


:subclass dissolve-wc
   process-step-descriptions
        ((FREE
              ((part-1 dissolve))
             ()
             ((acid (iron 10))
             ((foundry-worker 1))
        (FREE
        iron
        (:bulk purification iron-solution (iron 1.0))
        ))
:end-subclass dissolve-wc
```

```
:subclass cask-trash-wc
    process-step-descriptions
        ((FREE
            ((:bulk waste-container total-mass 100 nil (0.0 10) 100100))
            ()
            ()
            ((hse-worker 1))
        (FREE
        ()
        (:new waste-drum disposal-wo nil :default (pu 1.0))
        ))
:end-subclass cask-trash-wc

:subclass make-dilute-acid-solution
    workcenter-processing-time (0.0 0 8)
    process-step-descriptions
        ((FREE
            ()
            ()
            ((hno3 1)
             (water 100))
            ((chem-tech 1))
        (FREE
        hno3
        (:resource (dilute-acid hno3 101) (hno3 1.0)
        ))
:end-subclass make-dilute-acid-solution
```

9.0 Processes

Processes are a simplified syntax used to generate a location containing one workcenter and one bulk container. This workcenter will only process batches of material from that bulk container, and will only produce outputs that feed a bulk container at another location (thus a process can feed another process.)

This is an example of how a programmer can extend PWA to develop new classes of objects which use and interact with other objects within PWA.

Because processes are a simplified syntax, they have only some of the capabilities available from an explicitly defined location and workcenter. Users should be aware that processes cannot have constraints on the movement of material into them, or process parts or streams of material.

Consequently, should the user initially define a process and later discover that the equipment should be able to process parts, the process must be eliminated and a location/workcenter implementation be instituted. In such cases, it might be easier in the long-run to initially have implemented the processing as a workcenter rather than as a process.

9.1 Critieria for Identification of Processes

The user may define as many Processes as desired within a PWA simulation. A process may represent one piece of equipment, one workstation, or a set of equipment and workstations. The user determines how processes should be modeled for the simulation to provide meaningful results.

You may use the following criteria to determine which real-world objects should be in used in your simulation. A specific application of PWA may define additional criterion.

1. The process receives feed from only one bulk container and only produces bulk material outputs.
1. The types and amounts of material output by the process are of interest.
2. The types and amounts of material resources used by the process are of interest.
3. The process requires no reusable resources.
4. The time worked by the process is of interest.
5. The process represents a critical piece of equipment that may constrain production.

9.2 Process Attributes

The following characterisitcs (***attributes***) may be assigned to a process:

batch-size

Size of a normal batch of material. This is calculated based on the material specified in the batch-size-basis.

batch-size-basis

Name of the material to use in calculating the size of a batch. TOTAL-MASS may be used if the batch size is calculated on the total amount of material to be selected. On the other hand, if the primary material used is steel, you can use steel instead of TOTAL-MASS.

batch-minimum-size

Size of the smallest batch that will be processed.

batch-maximum-size

Size of the largest batch that will be processed.

materials-added-to-the-system

Material resources which will be added for the process.

batch-time

The time required to process a batch

output-primary-material

Name of the material to be used for default calculating the material output mass balances. E.G. If steel is the material output mass -- all other material amounts that are output will be proportionate to steel.

batch-outputs

A list of locations to feed material to by way of a material balance. The form of each feed specification on this list is:

(location-name bulk-container-name material-balance)

The material described by the material balance (see Workcenters) will be moved to the named location with a :feed output from the workcenter.

This is the only form of output available from a process.

initial-process-feeds

A list of material names and amounts which should be placed into the single bulk container which is associated with the process when the scenario is initialized

process-shift

The name of the shift which controls the times that the process operates.

process-shift-time

The amount of time needed in the shift for the process to be able to start.

Example

```
:class aluminum-processes
    :parents (processes)
        :instance dissolve
            initial-process-feed ((al 100))
            batch-size 5.0
            batch-minimum-size 1.0
            batch-size-basis total-mass
            batch-time (0.0 0 4)
            process-shift single-shift
            process-shift-time (0.0 0 4)
            materials-added-to-the-system
                ((acid 10.0) (catalyst 1.0))

            output-primary-material aluminum
            batch-outputs
                ((al-recovery al-solution (al .8) (acid .95) (catalyst 1.95))
                 (al-recovery al-sludge (al .2) (acid .05) (catalyst .05)))
```

10.0 Workorders

Parts move through a facility via a set of instructions that is attached to each part called a workorder. A workorder resembles a list of steps in sequence that indicates the places a part should go, the processing step(s) to be performed in each place and the duration of the processing step. Only parts move via a workorder.

Users create classes of workorders and each part will create its own copy (instance) of the appropriate workorder.

workorder-steps

A list of the sequence of processing steps. Each step is of the form:

(process-to-perform place time).

'TRANSFER' is a special workorder step to indicate a time delay when a part moves from one location to the next location. Movement of a part to a new location which requires no time delay can be specified by either a 'TRANSFER' of zero time, or by listing the next process and its location as the next step in the workorder, as is shown in the epoxy step of the following example.

Example

```
:class workorders
    :subclass part-preparation-wo
        workorder-steps
            ((transfer receiving-inspection (0.0 0 1))
             (cleanreceiving-inspection  (0.0 10))
             (dry   receiving-inspection (0.0 0 2))
             (epoxy  assembly-shop (0.0 0 2)))

:end-subclass part-preparation-wo
```

11.0 Random Deviates

Statistical distributions for generating random numbers in PWA are called Random Deviates. Several different distributions are available and programmers are encouraged to develop and extend the distributions as needed.

Users create specific distributions for use in a scenario by creating instances of the various types of distributions with specific parameters.

11.1 Exponential-Random-Deviate

Exponential-Random-Deviates will generate random numbers according to an exponential function with a user-specified mean.

mean

A number representing the mean of the distribution.

Example:

```
:class exponential-random-deviate
    :instance my-exponential-function
        mean 15
```

11.2 Uniform-Random-Deviate

Uniform-Random-Deviates will generate random numbers over a user-specified range.

low

A number representing the lowest number of the distribution.

high

A number representing the highest number of the distribution.

Example:

```
:class uniform-random-deviate
    :instance my-uniform-function
        low 50
        high 200
```

11.3 Normal-Random-Deviate

Normal-Random-Deviates will generate random numbers according to a normal distribution with a user-specified mean and standard deviation.

mean

A number representing the arithmatic mean of the distribution.

std-deviation

A number representing the standard deviation of the distribution (the average of how far it strays the mean).

Example:

```
:class normal-random-deviate
    :instance my-normal-function
        mean 200
        std-deviation 10
```

11.4 Constant-Random-Deviate

Constant-Random-Deviates will generate a user-specified constant. This is useful for testing scenarios.

constant

A number representing the constant value.

Example:

```
:class constant-random-deviate
    :instance my-constant-function
        constant 5
```

12.0 Running the Simulation

There are a number of mechanisms to run PWA to execute a scenario. The mechanisms are platform dependent.

12.1 SUN

12.1.1 Interactive

To run the model in an interactive mode, and to be able to stop the execution, inspect the model objects, and see the event trace as the model executes:

1. Open a shelltool.
2. Change directory to the directory which contains the scenario file(s). This is not necessary, but will enable the model to immediately find the scenario files.
3. Type tasp-V# where V# is the version of the model which should be executed.
4. The window in FIGURE 5. will pop up.

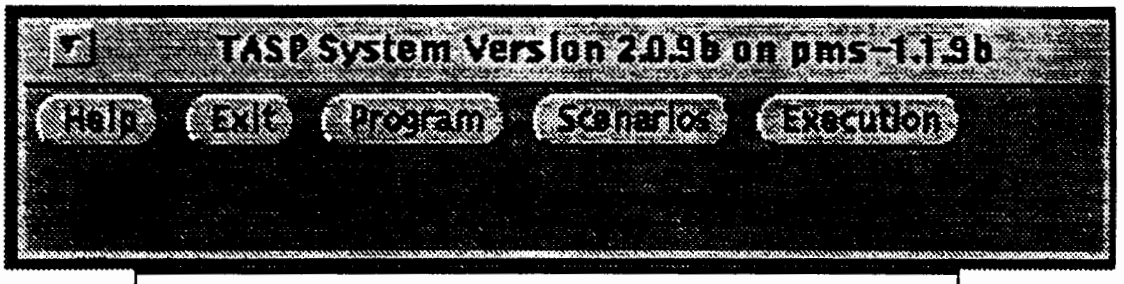


FIGURE 5.
PWA Main Interactive Window

5. Left-mouse on SCENARIOS and a window will pop up displaying the files in the directory. FIGURE 6. shows the Scenarios window. If the desired files are not displayed, the user may navigate the file system by:
   a. Left-mousing MOVE UP to go up one directory level,
   b. Selecting a directory which appears as one of the choices to move down into the directory,
   c. or left-mousing on the Directory line and typing the path to the desired directory.
6. Left-mouse on the file(s) to be used for the scenario. Files selected will be highlighted. If there are more files in the directory than can be displayed, the slider bar on the left side of the list may be used to move up or down within the directory.
7. Should a file be selected in error, left-mouse on it a second time to deselect it.
8. Left-mouse on INITIALIZE to initialize all highlighted files.

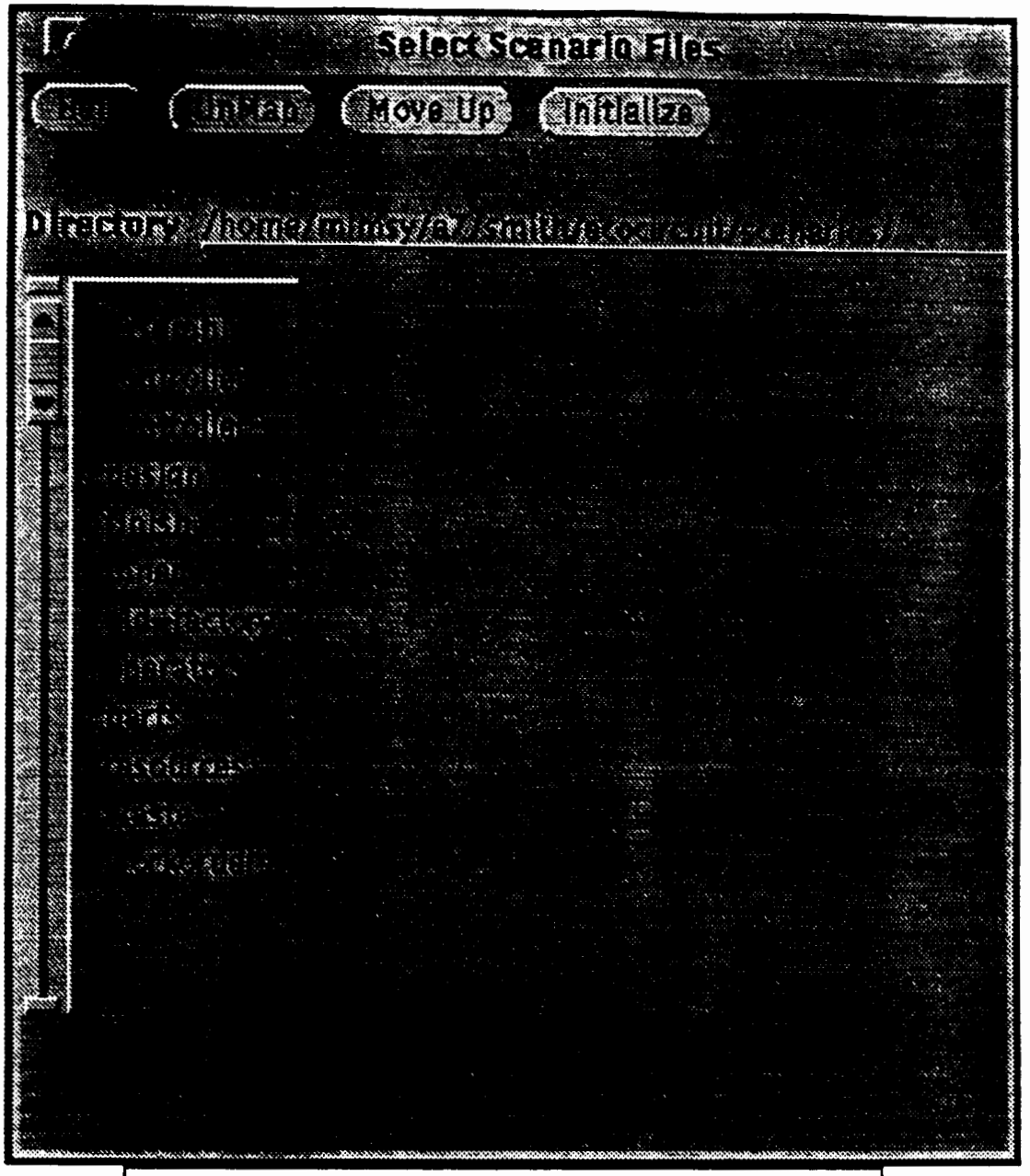


FIGURE 6.

PWA Scenario Selection Window

9. After the model has initialized, left-mouse on EXECUTION and a window will pop up. FIGURE 7. contains a sample Execution window.

10. Left-mouse on RUN to run the model, STEP to execute the events one at a time, and HALT to stop the running of the model.

The event trace is normally turned on. To turn it off:

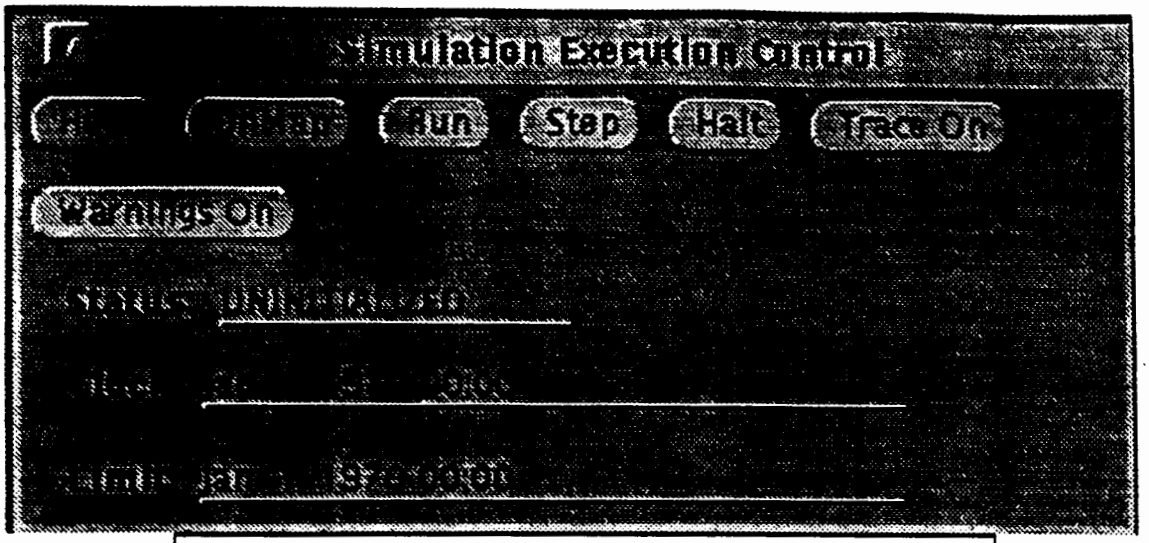


FIGURE 7.
PWA Simulation Execution Window

1. Left mouse on the TRACE ON button. The trace will be turned off and the button label will change to TRACE OFF to indicate that the trace is off.
2. To turn the trace on again, Left-mouse on TRACE OFF. The button label will change to TRACE ON to indicate that the trace is on.

The warning messages are normally turned on. Warnings inform the user if a complete mass balance was not specified, if a process-step-description was incomplete or if a number of other errors and potential problems were encountered. To turn the warnings off:

1. Left mouse on the WARNINGS ON button. The warnings will be turned off and the button label will change to WARNINGS OFF to indicate that no warnings will be printed.
2. To turn the warnings on again, Left-mouse on WARNINGS OFF. The button label will change to WARNINGS ON to indicate that warnings will be printed.

The PROGRAM button on the Main Window in FIGURE 5. will generate a mouse and menu interface for programmers to load, and compile.

1. Left-mouse on PROGRAM and a window similar to FIGURE 8. will pop up.
2. Load and Compile the source code files as needed.

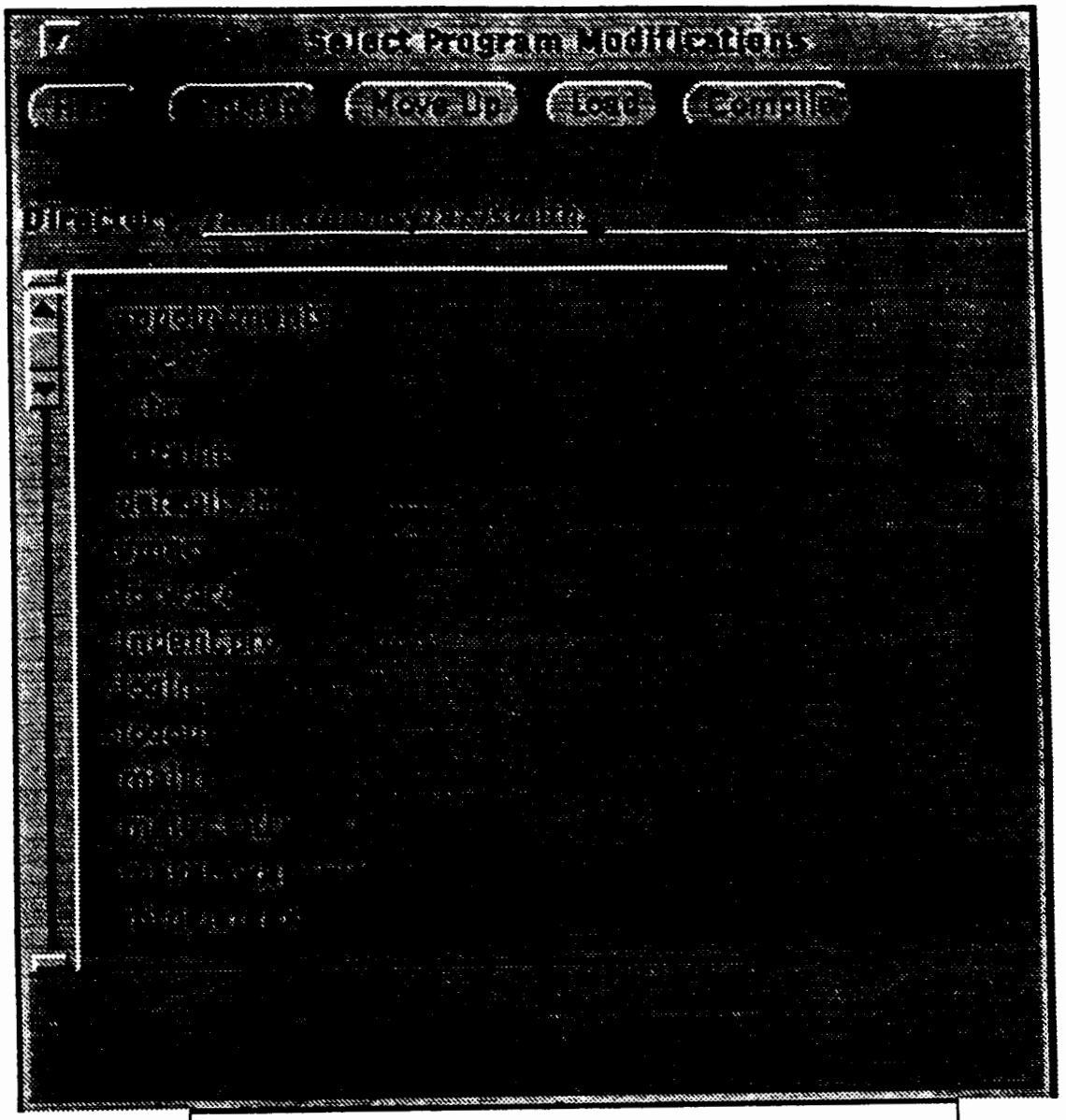


FIGURE 8.

PWA Program Modification Window

12.2 Batch Execution

The model can be run in a batch mode on the user's own workstation or can be submitted to a different workstation for execution.

12.2.1 Simple Local Batch

To run the model in batch mode on the user's own workstation, with one scenario file:

1. Open a shell tool
2. Change directory to the directory where the scenario files reside.
3. Type

```
tasp-V# -batch file-name-1 >file-name-2
```

where V# is the version of the TASP model which should be run.

file-name-1 is the name of the scenario file.

file-name-2 is the name of the file to which all warning messages should be written. Note: If the event trace is not turned off it will also be written to this file. To eliminate the writing of messages to a file and thus decrease run time, specify /dev/null as file-name-2.

Example

```
tasp-2.0.8b -batch my-scenario-file >/dev/null
```

12.2.2 Batch With Programmatic Control

To run the model in batch mode on the user's own workstation, with one or more scenario files turn off the event trace:

1. Create a file containing a script for a batch execution of the model. This file should contain:
   a. a statement that the LISP package is TASP,
   b. the simulation initialization should read a file(s),
   c. the event trace should be turned off (if this is the case),
   d. the warnings should be turned off (if this is the case), and
   e. a begin to start the simulation.

   Example:

```
(in-package "TASP")

(sim-init '("~/tasp-class/problem-2-loc"))

(trace-off)

(hush!)

(begin)
```

2. To run the model in batch mode on the user's own workstation -
   - Open a shell tool
   - Type

   tasp-V# -load file-name-1 >file-name-2

   where V# is the version of the TASP model.
   file-name-1 is the name of the file containing the load script developed in Step 1 and
   file-name-2 is the name of the file to which all warning messages should be written. Note: If the event trace is not turned off it will also be written to this file. To eliminate the writing of messages to a file and thus decrease run time, specify /dev/null as file-name-2.

Example:

```
tasp-2.0.8b -load my-load-script >my.log
```

3. To run the model in batch mode on a workstation other than the user's own workstation:

   a. Rlogin to the workstation on which the simulation should run.

   b. Type

   tasp-V# -load file-name-1 >file-name-2

   or

   tasp-V# -batch scenario >file-name-2

4. To run the model locally, or remote at a future time, or to submit a remote batch job and logoff the user's workstation:

   Type

   at time (CR)
   tasp-V# -load file-name-1 >file-name-2

   or

   at time (CR)
   tasp-V# -batch scenario >file-name-2

   where

   time is the time to start the job. This is a UNIX command and time may be specified in the forms:

   0815am Jan 24

   8:15am Jan 24

   now

   now + 1 day

   now + 1 hour

   5 pm Friday

   V# is the version of the TASP model which should be run.

   file-name-1 is the name of the file containing the load script developed in Step 1.

   file-name-2 is the name of the file to which all warning messages should be written. Note: If the event trace is not turned off it will also be written to this file. To eliminate the writing of messages to a file and thus decrease run time, specify /dev/null as file-name-2 or turn the warnings off in the script file.

   Example

   at 2000
   tasp-2.0.8b -load scenario-script.file >sim.log

13.0 Utilities for Debugging Scenarios

There are several functions available to assist in debugging a scenario. Unless otherwise noted, these commands are typed in the window which is running the model interactively.

(show)

Lists all the events on the event queue.

Example:

```
(show)
```

(describe (find-class 'object-name))

Prints all the information about the class.

Example:

```
(describe (find-class 'locations))
```

(describe (find-instance 'object-name))

Prints all the information about the instance.

Example:

```
(describe (find-instance 'receiving-dock))
```

(part-dump)

Prints a list of all parts with all the data for each part.

:verbose

When added to the scenario file, :verbose will print the initialization to the screen in interactive mode or to the log file in batch mode. The printing will start from the point that the :verbose is encountered in the scenario file.

Example:

```
:verbose
```

14.0 Auditors

The purpose of performing a simulation run is to develop data which will provide insights into the operation of a system. The collection of that data for later report generation and graphics display is the job of the accountants and auditors. The data are collected and summarized periodically, the timing of which is specified by the user.

audit-period

A keyword which specifies the periodicity of the collection of data. Allowed keywords are the keyword intervals specified in Section 2.1.2.

audit-base

The date for the first audit. The default is the beginning time of the simulation.

There are several different types of auditors, each designed to address the data which is applicable to each class within the model.

14.1 Audited-Locations

Add this as a parent to any location to collect audit data.

Example:

```
:class factory-locations
    :parents (locations audited-locations)
        audit-period :daily
        audit-base (0.0 0 0 1 1 1992)

        :instance room-a
```

14.2 Audited-Processes

Add this as a parent to any process to collect audit data.

Example:

```
:class factory-processes
    :parents (processes audited-processes)
    audit-period :daily
```

14.3 Audited-Operators

Add this as a parent to any operators to collect data.

Example:

```
:class techs
    :parents (operators audited-operators contact-exposed-operators)
        audit-period :weekly
        :instance chem-techs
            total-operator-resource 5
            contact-exposures (extremity penetrating)
```

14.4 Audited-Resources

Add this as a parent to any resource to collect audit data. Note: Operators have their own auditor to collect data which is specific to operators. See Audited-Operators.

Example:

```
:class factory-resources
    :parents (resources materials audited-resources)
        audit-period :yearly-calendar

        :instance di-water
```

14.5 Audited-Workcenters

Add this as a parent to any workcenters to collect audit data.

Example:

```
:class factory-workcenters
    :parents (mass-balance-workcenters audited-workcenters)
        audit-period :yearly-calendar
        :subclass lathes
        :end-subclass lathes
```

14.6 Audited-Parts

Add this as a parent to any class of parts to collect data.

Example:

```
:class factory-parts
    :parents (part audited-parts)
        audit-period :yearly-calendar

        :subclass ingots
        :end-subclass ingots
```

15.0 Foremen

Decision makers in the model are called Foremen. The user may create new foremen by mixing together the capabilities of several different foremen.

The user may cause events to occur within the simulation for Foremen. Events which may be directly posted are:

trigger

A command in the form

:at date-and-time (trigger nil nil)

will cause the foreman to respond to a trigger. This is used to force a foreman to perform his task.

Example:

```
:class PWA-reporter-foremen
        :instance report-generator
            :at (0.0 0 8 3 1 1992) (trigger nil nil)
```

15.1 Quitting-Foremen

These foremen halt the simulation when triggered.

Example:

```
:class factory-foremen
    :parents (quitting-foremen)
        :instance foreman-quitter
```

15.2 Warning-Foremen

These foremen will print a message to the user when triggered. The message will inform the user that a trigger alarm has occurred and what object caused the alarm.

Example:

```
:class factory-foremen
    :parents (warning-foremen)
        :instance foreman-warner
```

15.3 Debugging-Foremen

These foremen call the LISP debugger.

Example:

```
:class factory-foremen
    :parents (debugging-foremen)
        :instance foreman-for-programmers
```

15.4 Auditing-Foremen

These foremen force an audit on all audited objects in the model at the current time, regardless of the time of the last or next audit.

Example:

```
:class factory-foremen
    :parents (auditing-foremen)
        :instance foreman-auditor
```

15.5 PWA-Reporter-Foremen

These foremen generate the reports and other output files based on the data collected. The PWA-reporter foremen have a number of different types of reports which can be produced. In order to get a specific report, the user should list the report and the file into which the report should be written. Examples of each report are shown in Appendix C.

For each report enter full path name of the file. The paths and file names will be operating system specific.

processing-cumulative

Generates a table of the cumulative processing data.

processing-incremental

Generates a table of the incremental processing data.

feed

Generates a table showing the sources of all material into each location.

materials

Generates a table showing the material holdings for each location.

material-handling

Generates a table of the material inputs, additions, holdings and outputs for each location.

part-handling

Generates a table of parts handled by each location.

part-flow

Generates a table of the flow times of parts to each location.

ascii

Generates a file to be used as input to a graphics package or spreadsheet.

Special PWA REPORTS that can be generated in addition to the standard reports are as follows:

recra

A table of all of the rera listed materials

material-classes

A table of all the output materials by class

outfall

A table of all the outfall from every process

Example:

```
:class factory-foremen
    :parents (reporter-foremen)

    :instance reporter
        processing-cumulative "~/tasp/reports/proc-cum"
        processing-incremental "~/tasp/reports/proc-inc"
        feed          "~/tasp/reports/feed"
        materials  "~/tasp/reports/materials"
        material-handling "~/tasp/reports/material-handling"
        part-handling "~/tasp/reports/part-handling"
        part-flow  "~/tasp/reports/part-flow"
        ascii         "~/tasp/reports/ascii"
```

15.6 Equipment-Foremen

These foremen change the status of a specified workcenter when triggered.

target-workcenter

The name of the workcenter instance whose state should be changed. Note the workcenter must have an instance name.

triggered-status

The state in which the target workcenter should be placed.

Example:

```
:class factory-foremen
    :parents (equipment-foremen)
        :instance acid-maker-foreman
            target-workcenter acid-producing-wc-1
            triggered-status free
```

15.7 Mixed Foremen

It is possible to combine foremen capabilities. The following example will create a foreman that warns the user what is happening, halts the simulation, forces an audit to ensure the latest data are collected, and writes two reports.

Example:

```
:class factory-foremen
:parents (warning-foremen quitting-foremen auditing-foremen reporter-foremen)
        :instance reporter

        processing-cumulative "~/tasp/reports/proc-cum"
        processing-incremental "~/tasp/reports/proc-inc"
```

INTRODUCTION TO THE

PROCESS WASTE ASSESSMENT SYSTEM

Los Alamos

Course Objectives

Students shall be able to:

Define a real-world system in PWA terms, develop a model and run the simulation.

Develop and debug scenario input files.

Perform analysis using PWA

Use PWA as a potential tool to optimize waste management and waste minimization processes.

Los Alamos

USING PROCESS MODELING SYSTEM IN WASTE MANAGEMENT & MINIMIZATION

Los Alamos

Analysis Issues

The systems of a facility are dynamic.

There are many questions regarding effects of changes on the system.

Many questions are answered and operations can be improved by assessing the material processing.

- **Movement of material**
- **Processing of material**

Los Alamos

Addressing Analysis Issues

Most accurate assessment of effects is from changing the system.

Expensive and time-consuming to change the plant.

Expensive and time-consuming to build prototype facilities.

Cheaper and easier to build a simulation.

Los Alamos

Where Does PWA Apply?

PWA is a software tool designed to allow analysts, facility managers and others to describe the operations of their facility in terms of:

material movement from one place to another and

material transformations

Los Alamos

What is the Process Waste Assessment System?

Object-oriented discrete event computerized simulation tool for assisting waste coordinators to optimize operations through simulation analysis of their real-world processing environment

Inherent benefits to assist waste coordinators in tracking and managing waste in their facilities by developing and running a simulation model of their facility

Useful tool for waste management and minimization

Can be adapted to any operation and facility

Los Alamos

PWA ARCHITECTURE

Written in Common Lisp with the Common Lisp Object System (CLOS)

Set of libraries for analysts to describe constraints on movement and transformations of materials in terms applicable to the problem domain.

Set of software protocols for programmers to extend the simulation

Los Alamos

Current Hardware & Software Platforms

Sun Sparc

- SUN OS 4.1
- Lucid Common Lisp 4.0

Apple Macintosh

- System 7
- Apple MCL 2.0

IBM/PC

- MS-DOS 5.0
- Windows 3.1
- Franz Inc., Allegro CL/PC

Los Alamos

Support and Maintenance

Bug Reports

Software Distributions

Training

Upgrades

Los Alamos

Where has PWA been used?

Pantex	**Weapons returns/disassembly**
Ecocircuit	**Chemical waste treatment**
Rocky Flats	**Waste treatment of plutonium**
Complex 21	**Waste treatment of plutonium**
ARIES	**Experimental pu processing line**
Russian Plutonium Storage	**Movement & handling of nuclear materials**
LANL	**Process Waste Assessment**

Los Alamos

"WASTE IS A TERRIBLE THING TO MIND"

BUT

PWA IS A TOOL THAT CAN HELP THE WASTE COORDINATOR

Los Alamos

PWA HELPS WASTE COORDINATORS ANSWER THE FOLLOWING QUESTIONS:

WHAT COMPRISES WASTE IN MY FACILITY?

HOW DOES MY FACILITY GENERATE WASTE?

HOW CAN I TRACK AND MANAGE THE FLOW OF WASTE?

HOW CAN I TRACK AND MANAGE THE TRANSFORMATION OF WASTE AND ITS ASSOCIATED BY-PRODUCTS?

Los Alamos

QUESTIONS (cont.):

HOW CAN I IMPROVE THE PERFORMANCE OF MY WASTE TREATMENT AND STORAGE FACILITIES?

UNDER WHAT SET OF CONSTRAINTS, RESOURCES AND SCHEDULES CAN I ACHIEVE AN OPTIMAL WASTE MANAGEMENT & MINIMIZATION

Los Alamos

POTENTIAL APPLICATIONS OF

PWA

FOR WASTE MANAGEMENT & MINIMIZATION

Los Alamos

WASTE MINIMIZATION

METHODS THAT REDUCE OR ELIMINATE THE TOXICITY, MASS OR VOLUME OF WASTE:

METHODS THAT CAN BE MODELLED USING PWA

SUBSTITUTION

PROCESS MODIFICATION

SEGREGATION AND HOUSEKEEPING

REUSE AND RECYCLING

Los Alamos

PWA CAN BE USED AS A TOOL TO ADDRESS WASTE MANAGEMENT & MINIMIZATION PROBLEMS:

RADIOACTIVE LIQUID WASTE PIPELINES

LOW-LEVEL RADIOACTIVE SOLID WASTE

CHEMICAL, HAZARDOUS AND MIXED WASTE

TRANSURANIC (TRU) SOLID WASTE

Los Alamos

PWA CAN BE APPLIED TO THE FOLLOWING WASTE MANAGEMENT SITUATIONS:

TO ENSURE THE HANDLING, STORAGE & DISPOSAL OF ALL CLASSES OF WASTE

TO MODEL TOTALLY CHARACTERIZED WASTE STREAMS

TO ENSURE EFFECTIVE WASTE SEGREGATION

TO BETTER MANAGE WASTE PACKAGING LIMITATIONS & SPECIFICATIONS

Los Alamos

TO TRACK AND DISPOSE OF REACTIVE WASTE WITHIN ACUTE TIME SCHEDULES

TO MONITOR LEAKS & SPILLS AND MANAGE SPILL PREVENTION TECHNIQUES

INVENTORY CONTROL

and finally,

tracking and managing waste as it is generated!

Los Alamos

PWA

LIKE WASTE MANAGEMENT MAKES:

TECHNICAL SENSE

ECONOMIC SENSE

SOCIAL SENSE

Los Alamos

Process Waste Assessment System Description

Los Alamos

Purpose

PWA simulates the movement and processing of material.

Los Alamos

Material may be contained in:

Discrete Parts

Batches in Bulk containers

Continuous Streams

Stocks of Material Resources

Los Alamos

Movement

Materials move from one location to another location within a facility.

There may be constraints on movement.

Los Alamos

Processing

Processing may split, consolidate, or transform materials.

Processing must maintain a mass balance for all materials.
(The amount of material in the inputs = amount of material in the outputs)

Resources may be required for processing. (e.g. 5 units of machine oil required to lathe a 100 lb steel ingot)

One processing step may generate many products, residues, and wastes.

Los Alamos

Resources

There may be constraints on resources.

Resources may be materials or non-material items such as operators.

Los Alamos

Exposures

Personnel may be exposed to workplace hazards such as radiation, carcinogens, noise, and hazardous or toxic materials.

Los Alamos

Operating Schedules

Personnel or the facility in general may have one or more operating schedules which constrain production.

Los Alamos

Equipment Breakdown and Failures

Equipment may not always operate correctly.

Equipment breakdowns may require repair and may generate outputs other than the normal processing output.

Equipment failures do not require repair and may generate outputs other than the normal processing output.

Los Alamos

Processing Outputs

Waste and residue generation, handling, treatment and disposal is integrated with manufacturing production.

Los Alamos

How The PWA Works

Los Alamos

Concept

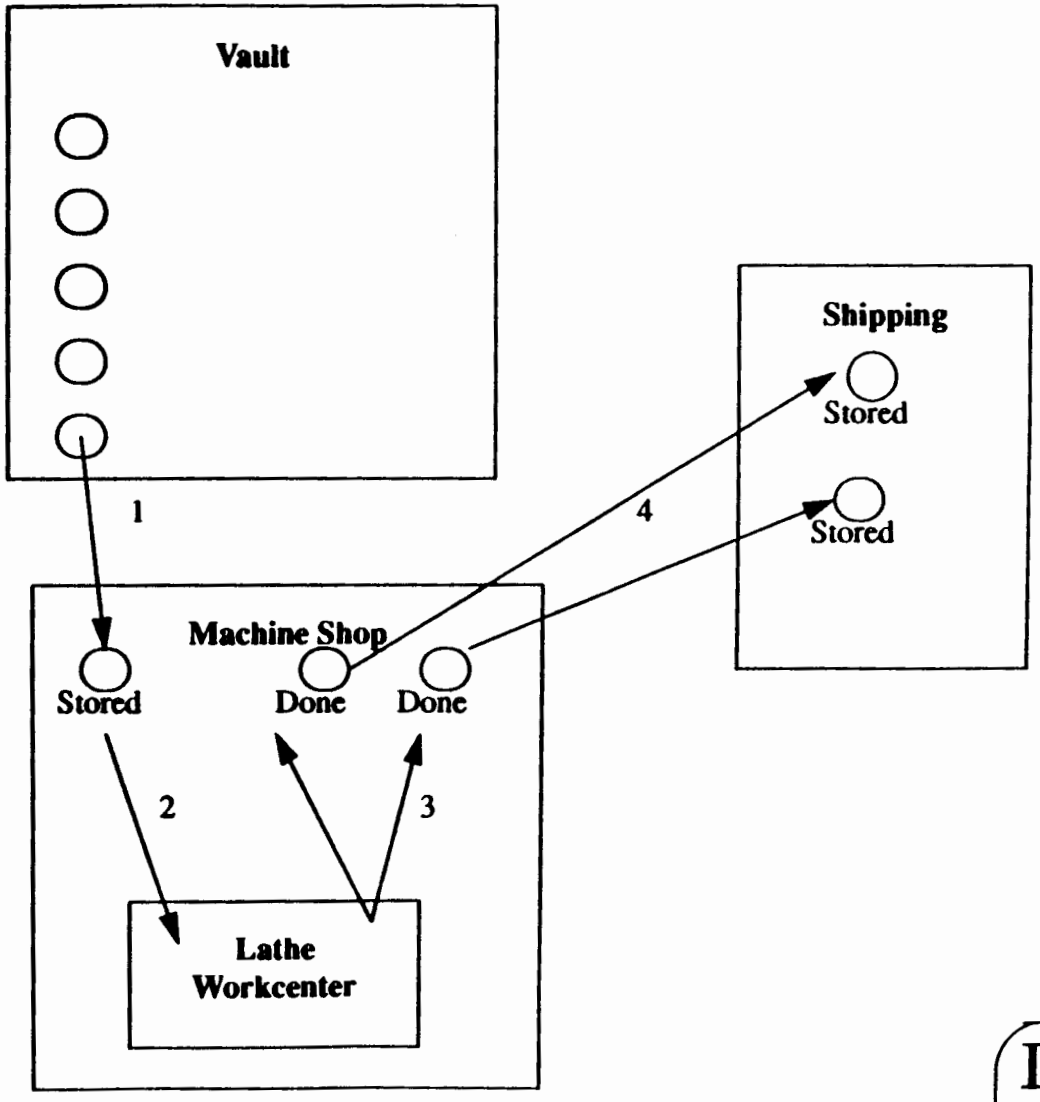
PWA is an object-oriented discrete-event simulation.

Events represent actions and decisions by objects within the model.

An event architecture describes causal relationship between events in the model.

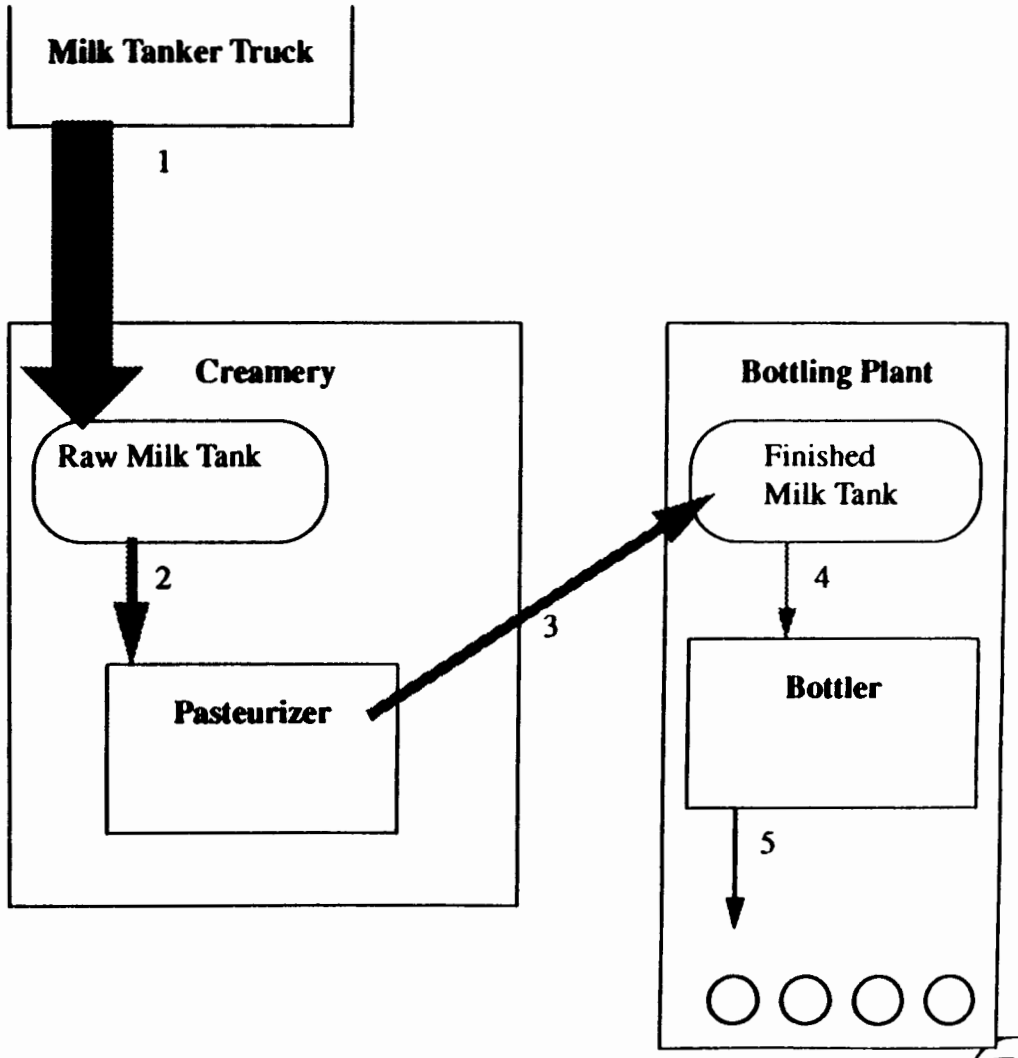
Los Alamos

Part Processing



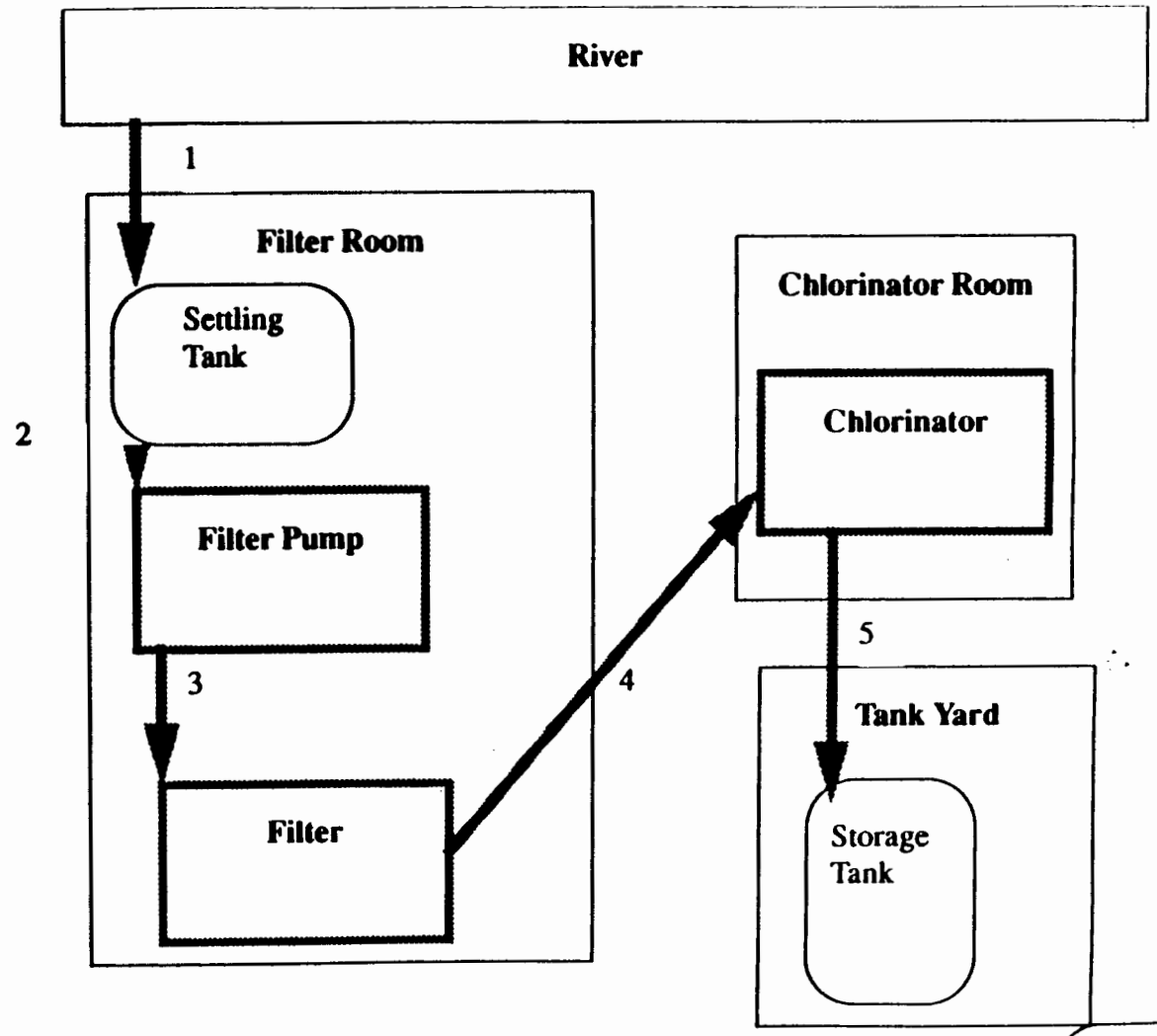
Los Alamos

Bulk Materials



Los Alamos

Streams



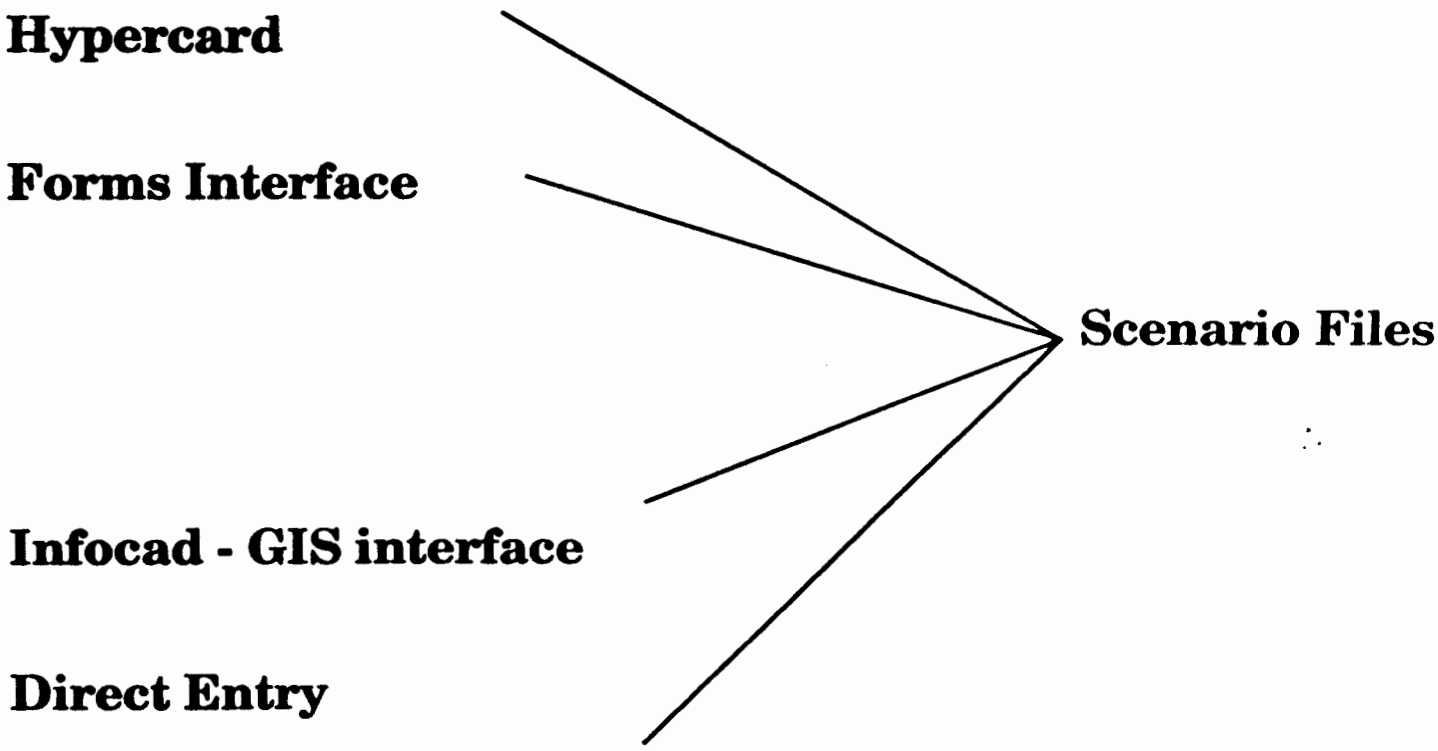
Los Alamos

Using the

Process Waste Assessment System

Los Alamos

Interfaces



Los Alamos

Scenario File

Real World Objects

Relationships of objects to each other and the interactions

Values of attributes of specific objects and related objects

Artifactual Objects

Objects and data for the simulation

- **Displays**
- **Stochastic Functions**
- **Reporting**
- **Simulation start and end dates**
- **Titles**

Los Alamos

Scenario Syntax

Mechanisms of Object-Oriented Programming

Classes

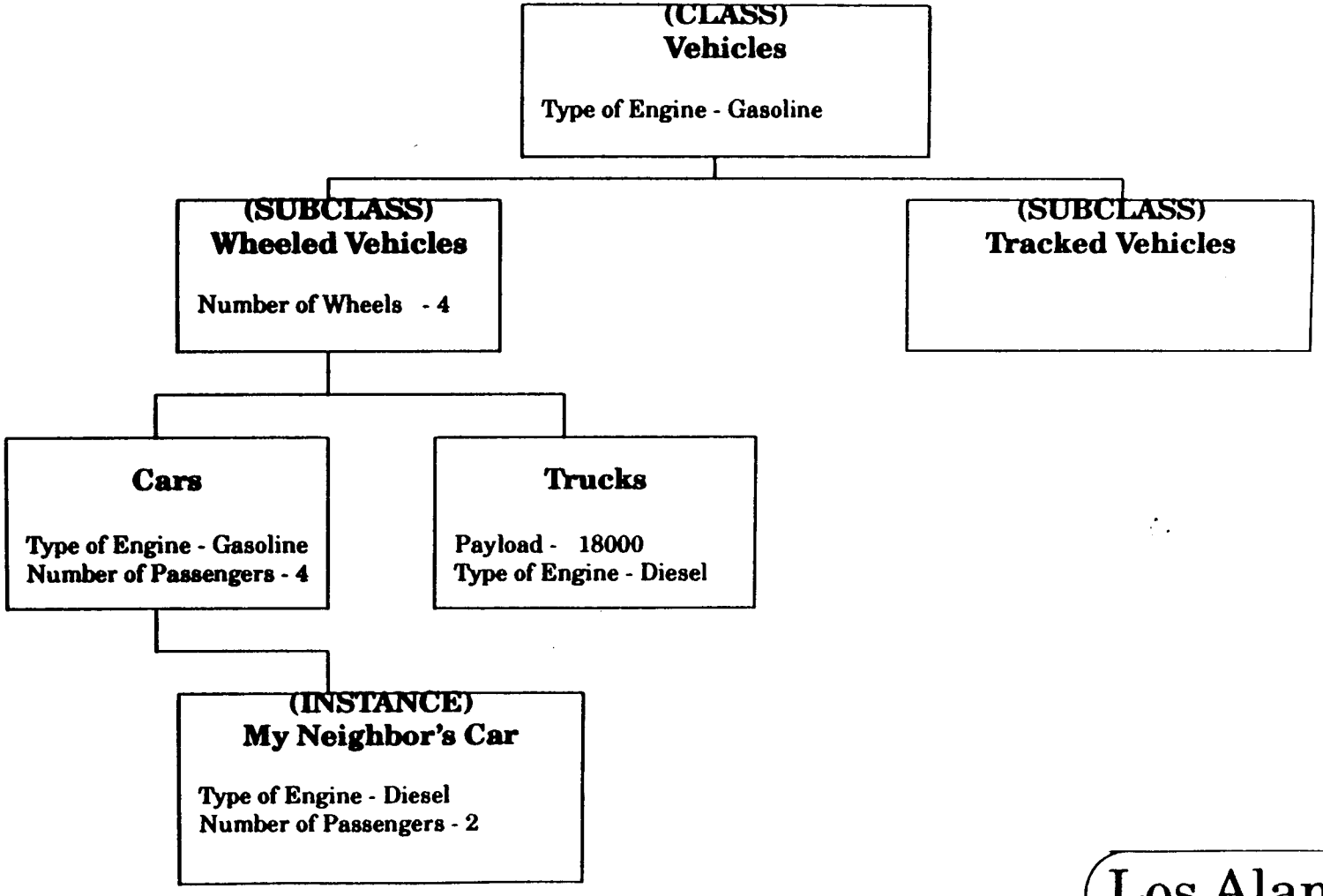
Subclasses

Instances

Is-a relationship

Los Alamos

Scenario Syntax
Classes and Inheritance



Los Alamos

Scenario Syntax Organization

Class-A

Instance-1

Attribute-1 x

Attribute-2 y

Instance-2

Attribute-1 x

Attribute-2 y

Class-A

Attribute-1 x

Attribute-2 y

Instance-1

Instance-2

Los Alamos

Scenario Syntax Organization

```
:class vehicles
    engine-type gasoline

:subclass wheeled-vehicles
    number-of-wheels  4

    :subclass cars
    number-of-passengers  4
        :instance my-neighbor's-car
        engine-type diesel
        number-of-passengers 2
    :end-subclass cars

:end-subclass wheeled-vehicles
```

Los Alamos

Scenario Syntax
Keywords

:class

:subclass

:end-subclass

:instance

:parent

:parents

:include

:title

:clock

:limit

:at

:print

<anything else>

Los Alamos

Scenario Syntax Example

```
:title "my-example-syntax"
:print "we are having fun now!"
:clock (0.0 0 0 1 1 1992)
:limit (0.0 0 0 1 1 1993)


:class LANL
:parents (DOE UC)
:subclass EM-Division
        :instance E-7
        :at current-simulation-clock go-home
        :instance rest-of-a-div
:end-subclass EM-Division
```

Los Alamos

Scenario Syntax
Times and Dates

Time and Date

(seconds minutes hours day month year) - (0.0 0 8 1 1 1992)

current-simulation-clock

Intervals

(seconds minutes hours day) - (0.0 0 8 1)

:daily

:weekly

:first-of-monthly

:last-of-monthly

:bi-monthly

:quarterly

:yearly-calendar

:yearly-fiscal

Los Alamos

Scenario Syntax
Comments

Use comments within a scenario file to

- **Document structure and data**
- **Assist others who may use the file**

Comments are preceded by semicolon

```
;; This is a comment
```

Comments may appear anywhere within a file

Los Alamos

MATERIALS AND MATERIAL CONTAINERS

Los Alamos

Materials

Represented as a pair in the form

(material-name material-amount)
(IRON 15) (WATER 25)

Material-names

User supplied names

No inherent knowledge assumed about materials

Material-amounts

Number to represent some amount of material

No units assumed

Los Alamos

Materials

Sets of material

((material-1 amount) (material-2 amount) (material-3 amount) ...)
((IRON 15) (NICKEL 25) (COBALT 10))

Sets of material move around the facility and are processed

Los Alamos

Material Containers

Sets of materials are held in material containers.

Parts

Bulk Containers

Streams

Material Resources

Los Alamos

Parts

Discrete chunks of material

The material is not divisible except as output from a processing step

Parts move around a facility by a workorder that is attached to each part.

Workorders specify:

- **where to move, processing step to perform, time for step**

Los Alamos

Parts

User defined classes of parts

Attributes -- (characteristics that further describe parts)

footprint	**storage space used**
part-status	**STORED or DONE**
workorder	**name of a type of workorder**
part-attributes	**list of attributes for part**
part-materials	**list of materials in the part**

Los Alamos

Parts Example

```
:class MY-PARTS
    :parents (parts)

        :subclass TYPE-1-PART

            part-materials ((nickel 1) (copper 10) (iron 15))
            workorder cut-apart-workorder
            part-attributes ((color blue) (designer acme))
            part-status stored
            footprint 2

        :end-subclass TYPE-1-PART

    :subclass TYPE-2-PART

            part-materials ((copper 10))
            workorder cut-apart-workorder

    :end-subclass TYPE-2-PART
```

Los Alamos

LOCATIONS

Los Alamos

Locations

Can assess the current state (decide if it can act upon a part)

Can create events appropriate to changes in the state

Can constrain movement of material

Can contain equipment and materials (workcenters & parts)

Los Alamos

Locations
Attributes

Attributes

stored-parts	list of parts initially in the location
stored-bulk	list of bulk containers in the location and initial materials in the containers
attached-workcenters	list of types of equipment in the location
exposures	list of types of exposures and amounts

Los Alamos

Locations
Attributes

Stored-parts

```
:class locations
    :instance building-3
        stored-parts ((ingots 20) (molds 40))
```

Stored-bulk

```
:class refinery
    :parents (locations)
    :instance building-3
        stored-bulk
            ((liquid-gunk)
             (initial-feed (h2o 2000) (glue 150))
             ((monitored-output output-tank)))
```

Los Alamos

Locations
Attributes

Attached-workcenters

```
:class locations
    :instance building-3
        attached-workcenters
            ((turret-lathe lathe-a)
             (turret-lathe lathe-b)
             drill-press)
```

Exposures

```
:class locations
    :instance building-3
        exposures ((noise 5) (penetrating-rad .01))
```

Los Alamos

Locations
Capabilities

Events that may be directly posted for locations:

receive-parts	**on a date receive a number of parts into the location**
decide-operation	**on a date decide what to do**

Los Alamos

Locations
Capabilities

receive-parts

```
:class warehouse
    :parents (locations)
        :instance receiving-dock
            :at (0.0 0 8 3 1 1992) (receive-parts (ingots 25))
```

decide-operation

```
:class warehouse
    :parents (locations)
        :instance receiving-dock
            :at (0.0 0 8 3 1 1992) decide-operation
```

Los Alamos

WORKCENTERS

Los Alamos

Workcenters

Any object that can process and/or transform material

Equipment

Workbench

Los Alamos

Workcenters

All PWA workcenters are MASS-BALANCE-WORKCENTERS

Users create subclasses of mass-balance-workcenters

- **Locations will create the appropriate instances**

Los Alamos

Workcenters Capabilities

Reside at a location

Attempt to process materials in the location

- **Examine workorders on parts**
- **Look in bulk containers**
- **Perform processes that have only resources as inputs**

Mass-balance-workcenters will maintain a strict mass balance

Los Alamos

Workcenters
Attributes

Attributes

workcenter-processing-time	default processing time when a workorder or batch time calculation is not present
exposures	list of types of exposures and amounts
breakdown-random-deviate	random-number generating object for breakdown time
repair-random-deviate	random-number generating object for repair time
failure-random-deviate	random-number generating object for number of operations to failure
process-step-description	list of processing behaviors of the equipment

Los Alamos

Workcenters

```
:class mass-balance-workcenters
    :subclass lathe
        workcenter-processing-time    (0.0 0 2)
        exposures                     ((penetrating-rad .05))
        breakdown-random-deviate      lathe-normal-breakdown
        repair-random-deviate         lathe-normal-repair
        failure-random-deviate        lathe-normal-failure
        process-step-descriptions     ()
    :end-subclass lathe
```

Los Alamos

Workcenters
Process-Step-Descriptions

A piece of equipment may be able to perform a number of different processes.

A piece of equipment can only perform one process at a time.

Each piece of equipment decides which process to perform by reading the process-step-descriptions and determining which starting conditions can be met.

Los Alamos

Workcenters
Process-Step-Descriptions

```
:class mass-balance-workcenters
    :subclass drill-presses
        process-step-descriptions
            (
            <description 1>
            <description 2>
            ...)
    :end-subclass drill-presses
```

Los Alamos

Workcenters

Some reasons a piece of equipment may not start:

A required input part is not in the location

A workorder step does not match the description

Insufficient material in a bulk container to start a batch

A material resource is not available

A bulk output cannot be received at the next location

Los Alamos

Workcenters
Process-Step-Descriptions

Initial Status

Input parts, stream, batches

( ) or nil

Material Resources

Reusable Resources

Normal Output

Failure Output

Breakdown Output

Los Alamos

Process-Step-Descriptions
Parts Simplest Case

Take a part add a material resource and output the part.

```
process-step-descriptions

        ((FREE                      ;; initial state
        ((part-1 add-water))        ;; select a part that needs water added
        ()                          ;; nil
        ((water 1.5))               ;; material added 1.5 units of water
        ()                          ;; no reusable resources needed

        (FREE                       ;; ending state for normal outputs
        nil                         ;; base material for mass balances
        (part-1)
        )
        )
```

Los Alamos

Process-Step-Descriptions
Parts Simple Case

Take a part add a material resource, use one operator, and split the part into two pieces.

```
process-step-descriptions

        ((FREE                                  ;; initial state
        ((part-1 add-water))                    ;; select a part that needs water added
        ()                                      ;; nil
        ((water 1.5))                           ;; material added 1.5 units of water
        ((operators 1))                         ;; 1 operator is needed

        (FREE                                   ;; ending state for normal outputs
        iron                                    ;; base material for mass balances
        (:new half-part part-1 part-1 (iron .5)) ;; first part to output
        (:new half-part part-1 part-1 (iron .5)) ;; second part to output
        )
        )
```

Los Alamos

Workorders

Los Alamos

Workorders

The processing sequence for parts.

Users create classes of workorders.

Los Alamos

Workorders

Attributes

workorder-steps	A list of the sequence of processing steps

Each workorder step is of the form

(process-to-perform place time)

TRANSFER is a special process-to-perform which moves a part from one location to another.

Los Alamos

Workorders

```
:class workorders

    :subclass part-preparation-wo

        workorder-steps
            ((transfer receiving-inspection (0.0 0 1))
             (clean    receiving-inspection (0.0 10))
             (dry      receiving-inspection (0.0 0 2))
             (epoxy    assembly-shop        (0.0 0 2)))

:end-subclass part-preparation-wo
```

Los Alamos

Problem 1

Los Alamos

Running the Simulation

Los Alamos

Interactive

Turn on the Macintosh

Select the Hard Disk icon

Select the PWA-1.0 icon

Los Alamos

(picture of Fred Window)

Select File (New) from the apple menu

this will automatically bring up an interactive FRED text editor window in which you will write your scenarios in Common Lisp and save them as *.Lisp* files.

Los Alamos

(Picture of Listener Window)

The Listener interactive window will also be on your screen This special Lisp window will automatically evaluate - line for line each line of code as you enter it. If you have make any errors in a given line of code, the Listener will tell you what is wrong and provide suggestions to make corrections.

Los Alamos

Use the Macintosh menus and shortcut keys to navigate around the common LISP environment just as you would for any Mac program.

ENTER THE LINES OF CODE THAT WILL COMPRISE YOUR SCENARIO FILE

UPON COMPLETION, SAVE YOUR FILE -- FILE (SAVE) FROM THE MAIN MENU

SELECT SCENARIO FROM THE SIMULATION MENU

SELECT YOUR FILE FROM THE LIST OF FILES IN THE DIALOG BOX

YOUR FILE WILL REAPPEAR IN A REFRESHED FRED WINDOW

SELECT RUN FROM THE SIMULATION MODEL

Los Alamos

YOU WILL SEE THE RESULTS OF YOUR SIMULATION IN THE LISTENER BOX.

YOU CAN THAN USE THE FILE (PRINT) OPTION TO PRINT OUT YOUR SIMULATION

Los Alamos

Debugging Scenarios

Functions to assist users:

show

describe

part-dump

:verbose

Los Alamos

Debugging Scenarios
Show

Lists all the events on the event queue

(show)

Los Alamos

Debugging Scenarios
Describe

Describe all the attributes of a class or an instance

(describe (find-class object-name))

(describe (find-class 'locations))

(describe (find-instance instance-name))

(describe (find-instance 'foundry))

Los Alamos

Debugging Scenarios
Part Dump

List all the parts and their attributes

(part-dump)

Los Alamos

Debugging Scenarios
:verbose

Prints the scenario back to the user during initialization.

Los Alamos

Bulk Containers

A discrete amount of material

The set of materials is divisible.

- **A subset of the materials may be selected for input to a process**

Bulk Containers do not move from location to location

- **but are assigned to a particular location**

Los Alamos

Bulk Containers

User defined classes of bulk containers

bulk-low-level-alarm	**material and low quantity at which an alarm should be triggered**
bulk-trigger-foreman	**name of a foreman to notify for an alarm**

Los Alamos

Bulk Containers

```
:class MY-TANKS
    :parents (bulk-containers)

        :subclass saline-tanks

            bulk-high-level-alarm 250
            bulk-low-level-alarm 25
            bulk-trigger-foreman saline-supply-foreman

        :end-subclass saline-tanks
```

Los Alamos

Streams

Continuous flow of material from one processing step to the next

Represents parallel processing

Pure transport mechanism

Moves material from one processing step to another

No user accessible attributes

Los Alamos

Process-Step-Descriptions
Batches Simple Case

Get a batch of material from a bulk container and make a part.

```
process-step-descriptions

        ((FREE
          ((:bulk molten-metal-vat iron 100 nil (0.0 10) 50 150))
          ()
          ()
          ((foundry-worker 1))
        (FREE
        iron
        (:new casting store-ingot-wo () (iron 1.0))
        ))
```

Los Alamos

Process-Step-Descriptions
Batches Simple Case

Get a batch of material from a bulk container, add a resource and output a batch of material.

process-step-descriptions

```
((FREE                                                    ;; initial status
  ((:bulk molten-metal-vat iron 100 nil (0.0 10) 50 150))  ;; batch spec
  ()                                                      ;; nil
  ((cobalt 50))                                           ;; material resource
  ()                                                      ;; reusable resource
(FREE                                                     ;; ending state
iron                                                      ;; primary material
(:feed foundry molten-alloy (iron 1.0))                   ;; batch output
))
```

Los Alamos

Process-Step-Descriptions
Streams Simple Case

Accept a stream of material, add a material resource and pass the stream on.

process-step-descriptions

```
((FREE                                                    ;; initial status
  ((:stream dissolved-material))                          ;; stream
  ()                                                      ;; nil
  ((salt 10))                                             ;; material resource
  ()                                                      ;; reusable resource
(FREE                                                     ;; ending state
iron                                                      ;; primary material
(:stream recovery-area dissolved-mat-and-salt (iron 1.0)) ;; stream output
))
```

Los Alamos

Material Resources

User defined materials

Material to be added to the simulation by a processing step

Material Resources come from a supply room or chemical stock

Los Alamos

Material Resources
Unconstrained Resources

Unlimited supply

Can contain only one material

Can only monitor material consumption

Los Alamos

Material Resources
Unconstrained Resources

:class FACTORY-RESOURCES

:parents (resources)

:instance cobalt
:instance water

Los Alamos

Material Resources
Materials

May generate user specified exposures

Attributes

exposures	**types and amounts of exposure**

Los Alamos

Material Resources
Materials

```
:class FACTORY-RESOURCES

    :parents (resources materials)

        :instance water

        :instance carbon-tetrachloride
            exposures ((carcinogen 1))

        :instance plutonium
            exposures ((penetrating-rad .05) (extremity-rad .01))
```

Los Alamos

Material Resources
Constrained Resources

Known finite amount of material available

May return a material other than the resource name

May return a set of materials

May trigger a decision by a foreman

Los Alamos

Material Resources
Materials

Attributes

available-resource	**Total amount that may be used**
resource-is-constraining	**T or nil to toggle constraint**
resource-material	**Material returned by a request**
extra-materials	**Trace elements returned with the material**
minimum-resource-level	**Amount below which an alarm will be triggered**
resource-trigger-foreman	**Name of a foreman to notify for an alarm**

Los Alamos

Material Resources
Constrained Resources

```
:class FACTORY-RESOURCES-CONSTRAINED
    :parents (resources-with-amounts)
        :instance nickel
            available-resource 50

        :instance impure-acid
            available-resource 100
            resource-is-constraining  T
            resource-material nitric-acid
            extra-materials ((iron 5) (nickel 2))
```

Los Alamos

Problem 2

Los Alamos

Auditors

Los Alamos

Auditors

Auditors collect data on the activities of objects within the model.

Auditors are added to objects in order to collect data.

In order to get data for reporting and post-processing, auditors must be used.

Los Alamos

Auditors

Attributes

Audit-period	**The periodicity of the audit. Interval keywords.**
Audit-base	**The time of the first audit.**

Los Alamos

Auditors

There are several types of auditors which may be used.

- **Audited-Locations**
- **Audited-Processes**
- **Audited-Operators**
- **Audited-Resources**
- **Audited-Workcenters**
- **Audited-Parts**

Los Alamos

Auditors

```
:class factory-locations
    :parents (locations audited-locations)
        audit-period :daily
        audit-base (0.0 0 0 1 1 1992)

        :instance room-a


:class factory-processes
    :parents (processes audited-processes)
        audit-period :daily
        audit-base (0.0 0 0 1 1 1992)
```

Los Alamos

Auditors

```
:class techs
    :parents (operators audited-operators contact-exposed-operators)
        audit-period :weekly
        :instance chem-techs
            total-operator-resource 5
            contact-exposures (extremity penetrating)


:class factory-resources
    :parents (resources materials audited-resources)
        audit-period :yearly-calendar

        :instance di-water


:class factory-workcenters
    :parents (mass-balance-workcenters audited-workcenters)
        audit-period :yearly-calendar
        :subclass lathes
        :end-subclass lathes
```

Los Alamos

Foremen

Los Alamos

Foremen

Decision Makers in the model are called Foremen.

Foremen act when triggered by some event within the simulation.

The triggers set on locations, bulk containers and resources will cause foremen to act.

Any foreman may be directly triggered by the scenario file.

Los Alamos

Foremen

There are several types of foremen.

Quitting-foremen	**Halts the simulation**
Warning-Foremen	**Prints a message to the user**
Debugging-Foremen	**Puts the simulation into the LISP debugger**
Auditing-Foremen	**Forces an audit for all audited objects**
Reporter-Foremen	**Prints user specified reports**
Equipment-Foremen	**Change the status of a piece of equipment**

Los Alamos

Foremen

Only Reporter-Foremen and Equipment-Foremen have attributes that may be set by users.

Los Alamos

Foremen
Reporter-Foremen

Users specify the reports that should be generated by defining which reports are wanted and the file names where the reports should be written.

The file names can be customized for each computing platform.

Los Alamos

Foremen
Reporter-Foremen

Attributes

- Processing-cumulative
- Processing-incremental
- Feed
- Materials
- Material-handling
- Part-handling
- Part-flow
- Ascii

Los Alamos

Foremen

```
:class reporter-foremen
        :instance report-generator
            :at (0.0 0 8 3 1 1992) (trigger nil nil)
```

```
:class factory-foremen
    :parents (quitting-foremen)
        :instance foreman-quitter
```

```
:class factory-foremen
    :parents (warning-foremen)
        :instance foreman-warner
```

Los Alamos

Foremen

```
:class factory-foremen
    :parents (auditing-foremen)
        :instance foreman-auditor


:class factory-foremen
    :parents (reporter-foremen)
        :instance reporter

        :at (0.0 0 8 3 1 1992) (trigger nil nil)

        processing-cumulative "pwa/reports/proc-cum"
        processing-incremental "pwa/reports/proc-inc"
        feed "pwa/reports/feed"
        materials"pwa/reports/materials"
        material-handling "pwa/reports/material-handling"
        part-handling "pwa/reports/part-handling"
        ascii "pwa/reports/ascii"
```

Los Alamos

Foremen

```
:class factory-foremen
:parents (warning-foremen quitting-foremen auditing-foremen reporter-foremen)
    :instance reporter
        processing-incremental "pwa/reports/proc-inc"
        feed "pwa/reports/feed"
```

Los Alamos

Problem 3

Los Alamos

Locations Constrained

unit-capacity-constrained-locations

capacity-constrained-locations

material-constrained-locations

maximal-footprint-locations

Los Alamos

Locations
Unit-Capacity-Constrained-Locations

Limits the number of parts in a location

Attributes

unit-capacity	number of parts that may be present
minimum-unit-trigger-level	minimum number of parts to notify foreman
maximum-unit-trigger-level	maximum number of parts to notify foreman
unit-trigger-foreman	foreman to notify for an alarm

Los Alamos

Locations
Unit-Capacity-Constrained-Locations

```
:class unit-capacity-constrained-locations
    :instance building-3

        unit-capacity 10

        minimum-unit-trigger-level 2

        maximum-unit-trigger-level 9

        unit-trigger-foreman building-3-warning-foreman
```

Los Alamos

Locations
Capacity-Constrained-Locations

Limits the total floor space available for parts

Attributes

storage-capacity	total floor space
minimum-capacity-trigger-level	minimum occupied space to notify foreman
maximum-capacity-trigger-level	maximum occupied to notify foreman
capacity-trigger-foreman	foreman to notify for an alarm

Los Alamos

Locations
Capacity-Constrained-Locations

:class capacity-constrained-locations

:instance building-3

storage-capacity 10

minimum-capacity-trigger-level 2

maximum-capacity-trigger-level 9

capacity-trigger-foreman building-3-warning-foreman

Los Alamos

Locations
Material-Constrained-Locations

Limits the amounts of a material or set of materials in the location

Attributes

critical-materials	list of material names and amounts

Los Alamos

Locations
Material-Constrained-Locations

```
:class nasty-places

    :parent material-constrained-locations

        critical-materials (
                                ((Pu U) 500)
                                 (acetone 50))
```

Los Alamos

Locations
Maximal-Footprint-Constrained-Locations

Limits the size of the largest part

Attributes

maximal-footprint	size of the largest part

Los Alamos

Locations
Maximal-Footprint-Constrained-Locations

:class small-places

:parent maximal-footprint-locations

maximal-footprint 2.5

Los Alamos

Locations
Mixing Types

Users can create new types of locations by mixing the constrained locations together.

```
:class warehouses
    :parent unit-capacity-constrained-locations
        unit-capacity 450

        :instance building-20
        :instance building-21
        :instance building-22
            unit-capacity 275


:class factories
    :parents (unit-capacity-constrained-locations capacity-constrained-locations)

        :instance assembly-building
                        100
            storage-capacity 200
```

Los Alamos

Reusable Resources

Los Alamos

Reusable Resources

Any user defined resource may be acquired by a workcenter as a reusable resource.

Los Alamos

Reusable Resources

Special classes of programmer defined resources

- **Working-shifts**
- **Operators**

Los Alamos

Reusable Resources
Working-Shifts

A time schedule which may be accessed by workcenters and operators.

Attributes

holidays	**A list of day month pairs when the shift would normally be available, but is not.**
shift-schedule	**A list of the start and end times for each shift for each day.**

Los Alamos

Reusable Resources
Working-Shifts

A time schedule which may be accessed by workcenters and operators.

Attributes

holidays	**A list of day month pairs when the shift would normally be available, but is not.**
shift-schedule	**A list of the start and end times for each shift for each day.**

Los Alamos

Reusable Resources
Working-Shifts

```
:class factory-shifts
    :parents (working-shifts)
        :instance one-shift-per-day
            shift-schedule (
             (((0.0 -30 0) (0.0 30 7))) ;; Monday
             (((0.0 -30 0) (0.0 30 7))) ;; Tuesday
             (((0.0 -30 0) (0.0 30 7))) ;; Wednesday
             (((0.0 -30 0) (0.0 30 7))) ;; Thursday
             (((0.0 -30 0) (0.0 30 7))) ;; Friday
             () ;;Saturday
             ()) ;; Sunday
            holidays ((1 1) (4 7) (25 12))
```

Los Alamos

Reusable Resources
Working-Shifts

```
:class factory-shifts
    :parents (working-shifts)
        :instance three-shifts
            shift-schedule (
             ( ((0.0 -30 0)  (0.0 30 7))      ;; Monday
               ((0.0   30 7)  (0.0 30 15))
               ((0.0   30 15) (0.0 30 23)) )
             ( ((0.0 -30 0)  (0.0 30 7))      ;; Tuesday
               ((0.0   30 7)  (0.0 30 15))
               ((0.0   30 15) (0.0 30 23)) )
             ( ((0.0 -30 0)  (0.0 30 7))      ;; Wednesday
               ((0.0   30 7)  (0.0 30 15))
               ((0.0   30 15) (0.0 30 23)) )
             ( ((0.0 -30 0)  (0.0 30 7))      ;; Thursday
               ((0.0   30 7)  (0.0 30 15))
               ((0.0   30 15) (0.0 30 23)) )
             ( ((0.0 -30 0)  (0.0 30 7))      ;; Friday
               ((0.0   30 7)  (0.0 30 15))
               ((0.0   30 15) (0.0 30 23)) )
             () ;;Saturday
             ()) ;; Sunday

            holidays ((1 1) (4 7) (25 12))
```

Los Alamos

Operators

Los Alamos

Operators

Operators are generally pools of operators and not individual operators.

For Example:

Apprentice Machinists

Journeyman Machinists

Electrical Technicians

EPA Auditors

Shift Supervisors

Los Alamos

Operators

Operators are reusable resources that do not become part of the output of a workcenter.

They can be:

Constrained

Assigned to shifts

Accrue exposures

Los Alamos

Operators

Unconstrained Operator Pool

```
:class techs
    :parents (operators)
        :instance chem-techs
```

Constrained Operators

```
:class techs
    :parents (operators)
        :instance chem-techs
            total-operator-resource 5
```

Los Alamos

Operators

Constrained Operators Assigned to Shifts

```
:class techs
    :parents (operators)
        :instance chem-techs
        total-operator-resource 5
        shift-assignment
            ((first-shift 3)
             (second-shift 2))
```

Los Alamos

Exposures

Los Alamos

Exposures

Pools of operators can accrue exposures.

If the average exposure in a pool exceeds the personnel limit for exposure, no operators from the pool will be available.

Los Alamos

Exposures

There are three main types of exposure:

Time Based:

- Location
- Workcenter

Contact

Non-contact

Los Alamos

Exposures
Time in a Location

Exposure accrued based on the number of operators in the location and the time spent there.

(number-of-operators * time-spent * exposure-emitted-by-location)

Los Alamos

Exposures
Time at a Workcenter

Exposure accrued based on the number of operators involved in the process and the time spent there.

(number-of-operators * time-spent * exposure-emitted-by-workcenter)

Los Alamos

Exposures
Contact

Exposure accrued based on the number of operators in contact with the material, the exposure emitted by the material, and the time spent there.**

(number-of-operators * time-spent * exposure-emitted-by-material)

**** Contact with material is defined as the material being processed.**

Los Alamos

Exposures
Non-contact

Exposure accrued based on the number of operators in the vicinity of the material, the exposure emitted by the material and the time spent.**

(number-of-operators * time-spent * exposure-emitted-by-material)

**** Non-contact with material is defined as the material in the room.**

Los Alamos

Workcenters
Initial Status

State for the workcenter for the description to apply

Default is FREE

Los Alamos

Workcenters
Inputs

Parts

or

Batches

or

Streams

or

()

Los Alamos

Workcenters
Input - Parts

A list of part-labels and processing-steps

Parts selected by processing step not by part class

All parts on the list must be in the location and available for processing

Los Alamos

Workcenters
Input - Parts

Process 1 part

((part1 drill-hole))

Process several parts

```
(
    (top      assemble-top)
    (bottom assemble-bottom)
    (side1   assemble-side)
    (side2   assemble-side)
    (side3   assemble-side)
    (side4   assemble-side)
)
```

Los Alamos

Workcenters
Input - Batches

A list of batches of materials to select

All the batches must be available in order to start processing

Los Alamos

Workcenters
Input - Batches

(:bulk container-name basis amount-descriptor amount time min max)

:bulk

container-name	**a bulk container in the location**
basis	**material name or TOTAL-MASS**
amount-descriptor	**batch size - number or (material fraction) or (TOTAL-MASS fraction)**
amount	**amount of material that can be processed in the time**
time	**time to process the amount**
min	**smallest allowable batch**
max	**largest allowable batch**

Los Alamos

Workcenters
Input - Batches

(:bulk tank TOTAL-MASS (TOTAL-MASS 0.5) 10 (0.0 30) 0 100)

((:bulk tank-1 iron (iron 25) 10 (0.0 30) 0 100)

(:bulk tank-2 acid (acid 50) 10 (0.0 30) 0 100))

Los Alamos

Workcenters
Input - Streams

The name of a stream from another workcenter

(:stream name)

(:stream anion-eluate)

Los Alamos

Workcenters
Material Resources

Resources that will become part of the mass balance calculation

(resource-amount amount-specification)

Get enough material to make the total of this material in the inputs equal the amount

(resource-amount :absolute amount-specification)

Add exactly this amount to the input

Los Alamos

Workcenters
Material Resources

The amount can be :

A number

(material-name multiplier)

(TOTAL-MASS multiplier)

(TOTAL-TIME multiplier)

Los Alamos

Workcenters
Material Resources

(oil 10)

(water (HNO3 .5))

(ni (TOTAL-MASS .5))

(acid (TOTAL-TIME 2))

Los Alamos

Workcenters
Material Resources

Combination of materials

A simple set of resources is combined by AND.

((water 10)
 (acid 5))

Los Alamos

Workcenters
Material Resources

Resource requests can contain logic

```
(OR (dirty-acid 10)
    (and (concentrated-acid 5)
         (water 5)))
```

Los Alamos

Workcenters
Reusable Resources

Requests same form as material resources

Reusable resources do not participate in the material balance

Represents

- **tools**
- **operators**
- **time**

Los Alamos

Workcenters
Outputs

Normal - output when equipment operates

Failure - output when equipment operates but output does not meet specification

Breakdown - output when equipment breaks

Los Alamos

Workcenters
Outputs

State

Default Material

Outputs

Los Alamos

Workcenters
Outputs - Status

State to leave the workcenter at the end of processing

Normally end state is FREE

Los Alamos

Workcenters
Outputs - Default Material

Material Name for material to use in defaulting rules

Los Alamos

Workcenters
Outputs

Pass an input part through

Make a new part

Send a stream

Pass a batch to a new location

Put a batch into a bulk-container in the same location

Make a resource

Los Alamos

Workcenters
Outputs - Parts

Passing an existing part

Part-label - from input parts list

Material-Balance - materials which go into the part

Los Alamos

Workcenters
Outputs - Parts

Create a new part

(:new part-type workorder attributes material-balance)

(:new fender fender-wo () MB)

(:new fender part-1 part-1 MB)

(:new fender fender-wo (part-1 (factory hill-stamping-plant)) MB)

Los Alamos

Workcenters
Outputs - Stream

(:stream Location Stream-Name Material-Balance)

(:stream catalytic-cracker crude-feed MB)

Los Alamos

Workcenters
Outputs - Bulk Container

Put a batch into a container at the same location or a different location

Put into a container at the same location - no explicit transfer

(:bulk Container-Name MB)

Put into a container at a different location or the same location with a transfer

(:feed Location Container-Name MB)

Los Alamos

Workcenters
Outputs - Bulk Container

(:bulk Container-Name MB)

(:bulk saline-tank MB)

(:feed Location Container-Name MB)

(:feed room-151 saline-tank MB)

Los Alamos

Workcenters
Outputs - Material Balance

Must specify all the materials

Apportioning materials is a three step process

Allocate absolute amounts

Allocate fractional amounts

Allocation using default rules.

Los Alamos

Workcenters
Outputs - Material Balance
Absolute Amounts

Put an absolute amount of material into an output

(material AMOUNT number)

material - name of the material

AMOUNT - indicates an absolute allocation

number - amount of material

Los Alamos

Workcenters
Outputs - Material Balance
Absolute Amounts

(pu AMOUNT 10)

Los Alamos

Workcenters
Outputs - Material Balance
Fractional Amounts

Apportion the remaining material according to the indicated fraction.

(material number)

material - name of material

number - percent of material to go into the output

Los Alamos

Workcenters
Outputs - Material Balance
Fractional Amounts

(material number)

(pu .5)

(fe 1.0)

Los Alamos

Workcenters
Outputs - Material Balance
Defaulting Rules

Default Material Rule

Default Output Rule

Errors

Los Alamos

Workcenters
Outputs - Material Balance
Default Materials

Identify a default material

All materials not specified by an absolute amount or fractional amount will follow the default material in proportion.

Los Alamos

Workcenters
Outputs - Material Balance
Default Materials

Input materials:

(Fe 100) (Ni 10) (C 6)

Output description:

```
(    Free
     Fe
     (:new half-ingot wo () (Fe 0.5))
     (:new half-ingot wo () (Fe 0.5)))
```

Each half-ingot will contain:

(Fe 50) (Ni 5) (C 3)

Los Alamos

Workcenters
Outputs - Material Balance
Default Output

Identify a default output

All materials not specified by an absolute amount or fractional amount will go into the default output

Los Alamos

Workcenters
Outputs - Material Balance
Default Output

Input materials:

(Fe 100) (Ni 10) (C 6)

Output description:

```
(
    Free
    nil
    (:new half-ingot wo () (Fe 0.49))
    (:new half-ingot wo () (Fe 0.49))
    (:bulk waste :default (Fe 0.02))
)
```

Each half-ingot will contain:

(Fe 49)

Waste will contain:

(Fe 2) (Ni 10) (C 6)

Los Alamos

Examples

```
:class test-factory
    :parents (mass-balance-workcenters)

    :subclass inspect-part-wc
        process-step-descriptions
            ((FREE
             ((part-1 inspect))
            ()
            ((water 1.5))
            ((inspectors 1))
            (FREE
            iron
            (part-1)
            ))
    :end-subclass inspect-part-wc
```

Los Alamos

Examples

```
:subclass weld-wc
    process-step-descriptions
    (
        (FREE
            ((part-1 weld-right)
            (part-2 weld-left))
            ()
            ((solder .25))
            ((certified-welders 1))
            (FREE
            iron
            (:new welded-part part-1part-1 (iron 1.0))
        )
        (FREE
            ((part-1 weld-iron-rod))
            ()
            ((solder .25)
            (iron .5))
            ((certified-welders 1))
            (FREE
            iron
            (:new welded-part part-1part-1 (iron 1.0))
        )
    )
```

Los Alamos

Examples

```
:subclass cut-apart-wc
    process-step-descriptions
        ((FREE
            ((part-1 cut))

            ((or (clean-oil .5)
                 (recycled-oil .75))
            ((certified-machinists 1))
        (FREE
        iron
        (:new half-part part-1part-1 (iron .49))
        (:new half-part part-1part-1 (iron .49))
        (:bulk scrap-turnings (iron .02))
        ))
:end-subclass cut-apart-wc
```

Los Alamos

Examples

```
:subclass cast-wc
    process-step-descriptions
        ((FREE
            ((:bulk molten-metal-vat iron 100 nil (0.0 10) 50 150))
           ()
           ()
           ((foundry-worker 1))
        (FREE
        iron
        (:new casting store-ingot-wo () (iron .98))
        (:feed foundry remelt-scrap (iron .02))
:end-subclass cast-wc
```

Los Alamos

```
Welcome to Macintosh Common Lisp Version 2.0!
?    Initialize from base scenario file Macintosh HD:Desktop Folder:Disk_PMS:EM_Problem3.
 lsp
:CLOCK "Jan  1, 1993 08:00"
:LIMIT "Jan  4, 1993 08:00"
:CLASS MY-FOREMEN
:PARENTS (REPORTER-FOREMEN AUDITING-FOREMEN)
:INSTANCE REPORT-GENERATOR
PROCESSING-CUMULATIVE "Cumulative-Problem-3"
PROCESSING-INCREMENTAL "Incremental-Problem-3"
FEED "feed--Problem-3"
MATERIALS "Materials--Problem-3"
MATERIAL-HANDLING "Material-hand-Problem-3"
PART-HANDLING "Part-Hand-Problem-3"
ASCII "ASCII-Problem-3"
:AT (0 0 7 4 1 1993) (TRIGGER NIL NIL)
:CLASS MY-PARTS
:PARENTS (PARTS)
:SUBCLASS RAW-PARTS
PART-MATERIALS ((STEEL 100))
WORKORDER LATHE-RAW-PARTS-WO
:END-SUBCLASS RAW-PARTS
:SUBCLASS LATHED-PARTS
:END-SUBCLASS LATHED-PARTS
:SUBCLASS BAG-OF-WASTE
:END-SUBCLASS BAG-OF-WASTE
:CLASS MY-FACTORY-CONSTRAINED-RESOURCES
:PARENTS (RESOURCES-WITH-AMOUNTS AUDITED-RESOURCES)
AUDIT-PERIOD :DAILY
 INSTANCE KIM-WIPES
AVAILABLE-RESOURCE 1000
:INSTANCE SOLVENT
AVAILABLE-RESOURCE 250
:CLASS MY-FACTORY-UNCONSTRAINED-RESOURCES
:PARENTS (RESOURCES AUDITED-RESOURCES)
AUDIT-PERIOD :DAILY
:INSTANCE MACHINE-OIL
:CLASS MY-LOCATIONS
:PARENTS (LOCATIONS AUDITED-LOCATIONS)
AUDIT-PERIOD :DAILY
:INSTANCE RECEIVING
STORED-PARTS ((RAW-PARTS 3 DONE))
:AT CURRENT-SIMULATION-CLOCK DECIDE-OPERATION
:INSTANCE LATHE-LOCATION
ATTACHED-WORKCENTERS ((LATHE-WC))
:INSTANCE SHIPPING-LOCATION
:INSTANCE WASTE-TREATMENT-LOCATION
STORED-BULK ((WASTE-TREATMENT-VAT) (CLEANED-STEEL-VAT))
ATTACHED-WORKCENTERS ((WASTE-TREATMENT-WC) (WASTE-TREATMENT-WC))
:INSTANCE WASTE-STORAGE-LOCATION
STORED-BULK ((WASTE-VAT))
:CLASS MY-WORKORDERS
:PARENTS (WORKORDERS)
:SUBCLASS LATHE-RAW-PARTS-WO
WORKORDER-STEPS ((TRANSFER LATHE-LOCATION (0 15)) (LATHE-A-PART LATHE-LOCATION (0 0 8)))
:END-SUBCLASS LATHE-RAW-PARTS-WO
:SUBCLASS MOVE-TO-SHIPPING-WO
WORKORDER-STEPS ((TRANSFER SHIPPING-LOCATION (0 15 0)))
:END-SUBCLASS MOVE-TO-SHIPPING-WO
:SUBCLASS MOVE-TO-WASTE-STORAGE-WO
WORKORDER-STEPS ((TRANSFER WASTE-STORAGE-LOCATION (0 0 2)))
:END-SUBCLASS MOVE-TO-WASTE-STORAGE-WO
:CLASS MY-MASS-BALANCE-WORKCENTERS
```

```
:PARENTS (MASS-BALANCE-WORKCENTERS AUDITED-WORKCENTERS)
AUDIT-PERIOD :DAILY
 SUBCLASS LATHE-WC
 ROCESS-STEP-DESCRIPTIONS ((FREE ((TEMP-PART LATHE-A-PART)) NIL ((MACHINE-OIL 5)) NIL (FR
EE STEEL (:NEW LATHED-PARTS MOVE-TO-SHIPPING-WO NIL (STEEL 0.99)) (:FEED WASTE-TREATMENT-
LOCATION WASTE-TREATMENT-VAT (STEEL 0.01) (MACHINE-OIL 1.0)))))
:END-SUBCLASS LATHE-WC
:SUBCLASS WASTE-TREATMENT-WC
PROCESS-STEP-DESCRIPTIONS ((FREE ((:BULK WASTE-TREATMENT-VAT STEEL 2 NIL (0 0 2) 2)) NIL
((KIM-WIPES 5) (SOLVENT 24)) NIL (FREE STEEL (:NEW BAG-OF-WASTE MOVE-TO-WASTE-STORAGE-WO
IL (KIM-WIPES 1.0) (SOLVENT 1.0) (MACHINE-OIL 1.0)) (:FEED WASTE-TREATMENT-LOCATION CLEANE
D-STEEL-VAT (STEEL 1.0)))))
:END-SUBCLASS WASTE-TREATMENT-WC
:EXIT
Create classes (VISIBLE-SIMULATION-OBJECTS)
Create instances (VISIBLE-SIMULATION-OBJECTS)
Initialize instances (VISIBLE-SIMULATION-OBJECTS)
Post initial events (VISIBLE-SIMULATION-OBJECTS)
Done
?
Jan  1, 1993 08:00 MACHINE-OIL  -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> DECIDE-OPERATION
Jan  1, 1993 08:00 KIM-WIPES    -> AUDIT-THYSELF
Jan  1, 1993 08:00 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> BEGIN-TRANSFER #<RAW-PARTS-3>
Jan  1, 1993 08:00 WASTE-TREATMENT-WC-3 -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> DECIDE-OPERATION
Jan  1, 1993 08:00 RECEIVING    -> BEGIN-TRANSFER #<RAW-PARTS-4>
Jan  1, 1993 08:00 RECEIVING    -> BEGIN-TRANSFER #<RAW-PARTS-5>
 an  1, 1993 08:00 RECEIVING    -> DECIDE-OPERATION
Jan  1, 1993 08:00 SOLVENT      -> AUDIT-THYSELF
Jan  1, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> AUDIT-THYSELF
Jan  1, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:00 LATHE-WC-3   -> AUDIT-THYSELF
Jan  1, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-3>
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-4>
Jan  1, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-5>
Jan  1, 1993 08:15 LATHE-WC-3   -> BEGIN-PROCESSING
Jan  1, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 LATHE-WC-3   -> END-PROCESSING
Jan  1, 1993 16:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-3> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  1, 1993 16:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 LATHE-WC-3   -> BEGIN-PROCESSING
Jan  1, 1993 16:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-5>
Jan  1, 1993 16:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-5>
Jan  1, 1993 16:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-WC-3   -> END-PROCESSING
Jan  2, 1993 00:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-3> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  2, 1993 00:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-WC-3   -> BEGIN-PROCESSING
Jan  2, 1993 00:15 WASTE-TREATMENT-WC-3 -> BEGIN-PROCESSING
```

```
Jan  2, 1993 00:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-6>
Jan  2, 1993 00:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-6>
Jan  2, 1993 00:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 02:15 WASTE-TREATMENT-WC-3 -> END-PROCESSING
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<WASTE-TREATMENT-WC-3> CLEAN
ED-STEEL-VAT ((STEEL 2.0))
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 02:15 WASTE-STORAGE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> BEGIN-TRANSFER #<BAG-OF-WASTE-7>
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 04:15 WASTE-STORAGE-LOCATION -> END-TRANSFER #<BAG-OF-WASTE-7>
Jan  2, 1993 04:15 WASTE-STORAGE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:00 KIM-WIPES     -> AUDIT-THYSELF
Jan  2, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-TREATMENT-WC-3 -> AUDIT-THYSELF
Jan  2, 1993 08:00 LATHE-WC-3    -> AUDIT-THYSELF
Jan  2, 1993 08:00 RECEIVING     -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 SOLVENT       -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 MACHINE-OIL   -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  2, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:15 LATHE-WC-3    -> END-PROCESSING
Jan  2, 1993 08:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-3> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  2, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-7>
Jan  2, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-7>
Jan  2, 1993 08:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  3, 1993 08:00 KIM-WIPES     -> AUDIT-THYSELF
Jan  3, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  3, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 MACHINE-OIL   -> AUDIT-THYSELF
Jan  3, 1993 08:00 RECEIVING     -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-WC-3 -> AUDIT-THYSELF
Jan  3, 1993 08:00 LATHE-WC-3    -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 SOLVENT       -> AUDIT-THYSELF
Jan  4, 1993 07:00 REPORT-GENERATOR -> TRIGGER NIL NIL
?
```

```
Welcome to Macintosh Common Lisp Version 2.0!
?     Initialize from base scenario file Macintosh HD:Desktop Folder:Disk_PMS:em_problem4. ?
 isp-
.CLOCK "Jan  1, 1993 08:00"
:LIMIT "Jan  4, 1993 08:00"
:CLASS MY-FOREMEN
:PARENTS (REPORTER-FOREMEN AUDITING-FOREMEN)
:INSTANCE REPORT-GENERATOR
PROCESSING-CUMULATIVE "Cumulative-Problem-4"
PROCESSING-INCREMENTAL "Incremental-Problem-4"
FEED "feed-Problem-4"
MATERIALS "Materials-Problem-4"
MATERIAL-HANDLING "Material-Report-Problem-4"
PART-HANDLING "Part-Hand-Problem-4"
ASCII "ASCII-Problem-4"
:AT (0 0 7 4 1 1993) (TRIGGER NIL NIL)
:CLASS MY-PARTS
:PARENTS (PARTS)
:SUBCLASS RAW-PARTS
PART-MATERIALS ((STEEL 100))
WORKORDER LATHE-RAW-PARTS-WO
:END-SUBCLASS RAW-PARTS
:SUBCLASS LATHED-PARTS
:END-SUBCLASS LATHED-PARTS
:SUBCLASS BAG-OF-WASTE
:END-SUBCLASS BAG-OF-WASTE
:SUBCLASS COMPACTED-STEEL-PARTS
:END-SUBCLASS COMPACTED-STEEL-PARTS
:CLASS MY-FACTORY-CONSTRAINED-RESOURCES
PARENTS (RESOURCES-WITH-AMOUNTS AUDITED-RESOURCES)
AUDIT-PERIOD :DAILY
:INSTANCE KIM-WIPES
AVAILABLE-RESOURCE 1000
:INSTANCE SOLVENT
AVAILABLE-RESOURCE 250
:CLASS MY-FACTORY-RESOURCES
:PARENTS (RESOURCES AUDITED-RESOURCES)
AUDIT-PERIOD :DAILY
:INSTANCE MACHINE-OIL
:CLASS MY-LOCATIONS
:PARENTS (LOCATIONS AUDITED-LOCATIONS)
AUDIT-PERIOD :DAILY
:INSTANCE RECEIVING
STORED-PARTS ((RAW-PARTS 3 DONE))
:AT CURRENT-SIMULATION-CLOCK DECIDE-OPERATION
:INSTANCE LATHE-LOCATION
ATTACHED-WORKCENTERS ((LATHE-WC))
:INSTANCE SHIPPING-LOCATION
:INSTANCE WASTE-TREATMENT-LOCATION
STORED-BULK ((WASTE-TREATMENT-VAT) (CLEANED-STEEL-VAT))
ATTACHED-WORKCENTERS (WASTE-TREATMENT-WC WASTE-TREATMENT-WC COMPACTING-WC)
:INSTANCE WASTE-STORAGE-LOCATION
:CLASS MY-WORKORDERS
:PARENTS (WORKORDERS)
:SUBCLASS LATHE-RAW-PARTS-WO
WORKORDER-STEPS ((TRANSFER LATHE-LOCATION (0 15)) (LATHE-A-PART LATHE-LOCATION (0 0 8)))
:END-SUBCLASS LATHE-RAW-PARTS-WO
:SUBCLASS MOVE-TO-SHIPPING-WO
WORKORDER-STEPS ((TRANSFER SHIPPING-LOCATION (0 15 0)))
:END-SUBCLASS MOVE-TO-SHIPPING-WO
:SUBCLASS MOVE-TO-WASTE-STORAGE-WO
WORKORDER-STEPS ((TRANSFER WASTE-STORAGE-LOCATION (0 0 2)))
:END-SUBCLASS MOVE-TO-WASTE-STORAGE-WO
```

```
:SUBCLASS MOVE-TO-LATHE-LOCATION-WO
WORKORDER-STEPS ((TRANSFER LATHE-LOCATION (0 0 2)))
:END-SUBCLASS MOVE-TO-LATHE-LOCATION-WO
:CLASS MY-MASS-BALANCE-WORKCENTERS
:PARENTS (MASS-BALANCE-WORKCENTERS AUDITED-WORKCENTERS)
AUDIT-PERIOD :DAILY
:SUBCLASS LATHE-WC
PROCESS-STEP-DESCRIPTIONS ((FREE ((TEMP-PART LATHE-A-PART)) NIL ((MACHINE-OIL 5)) NIL (FR
%EE STEEL (:NEW LATHED-PARTS MOVE-TO-SHIPPING-WO NIL (STEEL 0.99)) (:FEED WASTE-TREATMENT-
%OCATION WASTE-TREATMENT-VAT (STEEL 0.01) (MACHINE-OIL 1.0)))))
:END-SUBCLASS LATHE-WC
:SUBCLASS WASTE-TREATMENT-WC
PROCESS-STEP-DESCRIPTIONS ((FREE ((:BULK WASTE-TREATMENT-VAT STEEL 2 NIL (0 0 2) 2)) NIL
%((KIM-WIPES 5) (SOLVENT 24)) NIL (FREE STEEL (:NEW BAG-OF-WASTE MOVE-TO-WASTE-STORAGE-WO N
%IL (KIM-WIPES 1.0) (SOLVENT 1.0) (MACHINE-OIL 1.0)) (:BULK CLEANED-STEEL-VAT (STEEL 1.0)))
%))
:END-SUBCLASS WASTE-TREATMENT-WC
:SUBCLASS COMPACTING-WC
PROCESS-STEP-DESCRIPTIONS ((FREE ((:BULK CLEANED-STEEL-VAT STEEL 1.0 NIL (0 0 2) 1.0)) NI
%L NIL NIL (FREE NIL (:NEW COMPACTED-STEEL-PARTS MOVE-TO-LATHE-LOCATION-WO NIL (STEEL 1.0))
%)))
:END-SUBCLASS COMPACTING-WC
:EXIT
Create classes (VISIBLE-SIMULATION-OBJECTS)
Create instances (VISIBLE-SIMULATION-OBJECTS)
Initialize instances (VISIBLE-SIMULATION-OBJECTS)
Post initial events (VISIBLE-SIMULATION-OBJECTS)
Done
?
Jan  1, 1993 08:00 MACHINE-OIL  -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> DECIDE-OPERATION
Jan  1, 1993 08:00 KIM-WIPES    -> AUDIT-THYSELF
Jan  1, 1993 08:00 COMPACTING-WC-2 -> AUDIT-THYSELF
Jan  1, 1993 08:00 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> BEGIN-TRANSFER #<RAW-PARTS-2>
Jan  1, 1993 08:00 RECEIVING    -> AUDIT-THYSELF
Jan  1, 1993 08:00 RECEIVING    -> DECIDE-OPERATION
Jan  1, 1993 08:00 RECEIVING    -> BEGIN-TRANSFER #<RAW-PARTS-4>
Jan  1, 1993 08:00 RECEIVING    -> BEGIN-TRANSFER #<RAW-PARTS-3>
Jan  1, 1993 08:00 RECEIVING    -> DECIDE-OPERATION
Jan  1, 1993 08:00 SOLVENT      -> AUDIT-THYSELF
Jan  1, 1993 08:00 WASTE-TREATMENT-WC-1 -> AUDIT-THYSELF
Jan  1, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:00 LATHE-WC-2   -> AUDIT-THYSELF
Jan  1, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-2>
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-3>
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-4>
Jan  1, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 08:15 LATHE-WC-2   -> BEGIN-PROCESSING
Jan  1, 1993 16:15 LATHE-WC-2   -> END-PROCESSING
Jan  1, 1993 16:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-2> WASTE-TREATMENT
%-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  1, 1993 16:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 LATHE-WC-2   -> BEGIN-PROCESSING
Jan  1, 1993 16:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-4>
Jan  1, 1993 16:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-4>
```

```
Jan  1, 1993 16:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-WC-2    -> END-PROCESSING
Jan  2, 1993 00:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-2> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  2, 1993 00:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-WC-2    -> BEGIN-PROCESSING
Jan  2, 1993 00:15 WASTE-TREATMENT-WC-2 -> BEGIN-PROCESSING
Jan  2, 1993 00:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-5>
Jan  2, 1993 00:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-5>
Jan  2, 1993 00:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 02:15 WASTE-TREATMENT-WC-2 -> END-PROCESSING
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 02:15 WASTE-STORAGE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 02:15 COMPACTING-WC-2 -> BEGIN-PROCESSING
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> BEGIN-TRANSFER #<BAG-OF-WASTE-6>
Jan  2, 1993 02:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 04:15 COMPACTING-WC-2 -> END-PROCESSING
Jan  2, 1993 04:15 WASTE-STORAGE-LOCATION -> END-TRANSFER #<BAG-OF-WASTE-6>
Jan  2, 1993 04:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 04:15 WASTE-STORAGE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 04:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 04:15 COMPACTING-WC-2 -> BEGIN-PROCESSING
Jan  2, 1993 04:15 WASTE-TREATMENT-LOCATION -> BEGIN-TRANSFER #<COMPACTED-STEEL-PARTS-6>
Jan  2, 1993 04:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 06:15 COMPACTING-WC-2 -> END-PROCESSING
Jan  2, 1993 06:15 LATHE-LOCATION -> END-TRANSFER #<COMPACTED-STEEL-PARTS-6>
Jan  2, 1993 06:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 06:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 06:15 WASTE-TREATMENT-LOCATION -> BEGIN-TRANSFER #<COMPACTED-STEEL-PARTS-7>
Jan  2, 1993 06:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  2, 1993 08:00 MACHINE-OIL   -> AUDIT-THYSELF
Jan  2, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 COMPACTING-WC-2 -> AUDIT-THYSELF
Jan  2, 1993 08:00 KIM-WIPES     -> AUDIT-THYSELF
Jan  2, 1993 08:00 LATHE-WC-2    -> AUDIT-THYSELF
Jan  2, 1993 08:00 SOLVENT       -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
Jan  2, 1993 08:00 WASTE-TREATMENT-WC-1 -> AUDIT-THYSELF
Jan  2, 1993 08:00 RECEIVING     -> AUDIT-THYSELF
Jan  2, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<COMPACTED-STEEL-PARTS-7>
Jan  2, 1993 08:15 LATHE-WC-2    -> END-PROCESSING
Jan  2, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-2> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  2, 1993 08:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-6>
Jan  2, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-6>
Jan  2, 1993 08:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  3, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-WC-1 -> AUDIT-THYSELF
Jan  3, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 MACHINE-OIL   -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  3, 1993 08:00 RECEIVING     -> AUDIT-THYSELF
Jan  3, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
```

```
Jan  3, 1993 08:00 KIM-WIPES     -> AUDIT-THYSELF
Jan  3, 1993 08:00 COMPACTING-WC-2 -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 LATHE-WC-2    -> AUDIT-THYSELF
Jan  3, 1993 08:00 SOLVENT       -> AUDIT-THYSELF
Jan  4, 1993 07:00 REPORT-GENERATOR -> TRIGGER NIL NIL
?
```

```
Welcome to Macintosh Common Lisp Version 2.0!
?    Initialize from base scenario file Macintosh HD:Desktop Folder:Disk_PMS:em_problem5.
lisp
.CLOCK "Jan  1, 1993 08:00"
:LIMIT "Jan  4, 1993 08:00"
:CLASS MY-FOREMEN
:PARENTS (REPORTER-FOREMEN AUDITING-FOREMEN)
:INSTANCE REPORT-GENERATOR
PROCESSING-CUMULATIVE "Cumulative-Problem-5"
PROCESSING-INCREMENTAL "Incremental-Problem-5"
FEED "feed-Problem-5"
MATERIALS "Materials-Problem-5"
MATERIAL-HANDLING "Material-Hand-Problem-5"
PART-HANDLING "Part-Hand-Problem-5"
ASCII "ASCII-Problem-5"
:AT (0 0 7 4 1 1993) (TRIGGER NIL NIL)
:CLASS MY-PARTS
:PARENTS (PARTS)
:SUBCLASS RAW-PARTS
PART-MATERIALS ((STEEL 100))
WORKORDER LATHE-RAW-PARTS-WO
:END-SUBCLASS RAW-PARTS
:SUBCLASS LATHED-PARTS
:END-SUBCLASS LATHED-PARTS
:SUBCLASS LATHED-COMPACTED-PARTS
:END-SUBCLASS LATHED-COMPACTED-PARTS
:SUBCLASS BAG-OF-WASTE
:END-SUBCLASS BAG-OF-WASTE
:SUBCLASS COMPACTED-STEEL-PARTS
:END-SUBCLASS COMPACTED-STEEL-PARTS
:CLASS MY-FACTORY-RESOURCES
:PARENTS (RESOURCES AUDITED-RESOURCES)
AUDIT-PERIOD :DAILY
:INSTANCE MACHINE-OIL
:CLASS MY-FACTORY-CONSTRAINED-RESOURCES
:PARENTS (RESOURCES-WITH-AMOUNTS AUDITED-RESOURCES)
AUDIT-PERIOD :DAILY
:INSTANCE KIM-WIPES
AVAILABLE-RESOURCE 1000
:INSTANCE SOLVENT
AVAILABLE-RESOURCE 250
:CLASS MY-LOCATIONS
:PARENTS (LOCATIONS AUDITED-LOCATIONS)
AUDIT-PERIOD :DAILY
:INSTANCE RECEIVING-LOCATION
STORED-PARTS ((RAW-PARTS 3 DONE))
:AT CURRENT-SIMULATION-CLOCK DECIDE-OPERATION
:INSTANCE LATHE-LOCATION
ATTACHED-WORKCENTERS ((LATHE-WC))
:INSTANCE SHIPPING-LOCATION
:INSTANCE WASTE-TREATMENT-LOCATION
STORED-BULK ((WASTE-TREATMENT-VAT) (CLEANED-STEEL-VAT))
ATTACHED-WORKCENTERS (WASTE-TREATMENT-WC WASTE-TREATMENT-WC COMPACTING-WC)
:INSTANCE WASTE-STORAGE-LOCATION
:CLASS MY-WORKORDERS
:PARENTS (WORKORDERS)
:SUBCLASS LATHE-RAW-PARTS-WO
WORKORDER-STEPS ((TRANSFER LATHE-LOCATION (0 15)) (LATHE-A-RAW-PART LATHE-LOCATION (0 0 8
))
:END-SUBCLASS LATHE-RAW-PARTS-WO
:SUBCLASS MOVE-TO-SHIPPING-WO
WORKORDER-STEPS ((TRANSFER SHIPPING-LOCATION (0 15 0)))
:END-SUBCLASS MOVE-TO-SHIPPING-WO
```

```
Jan  2, 1993 08:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:45 LATHE-WC-2    -> END-PROCESSING
Jan  2, 1993 08:45 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-2> WASTE-TREATMENT
-VAT ((MACHINE-OIL 10.0) (STEEL 0.01))
Jan  2, 1993 08:45 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:45 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:45 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:45 LATHE-WC-2    -> BEGIN-PROCESSING
Jan  2, 1993 08:45 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-COMPACTED-PARTS-7>
Jan  2, 1993 08:45 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 09:00 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-COMPACTED-PARTS-7>
Jan  2, 1993 09:00 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 09:15 LATHE-WC-2    -> END-PROCESSING
Jan  2, 1993 09:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-2> WASTE-TREATMENT
-VAT ((MACHINE-OIL 10.0) (STEEL 0.01))
Jan  2, 1993 09:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 09:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 09:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 09:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-COMPACTED-PARTS-8>
Jan  2, 1993 09:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 09:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-COMPACTED-PARTS-8>
Jan  2, 1993 09:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  3, 1993 08:00 WASTE-TREATMENT-WC-1 -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-STORAGE-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 MACHINE-OIL  -> AUDIT-THYSELF
Jan  3, 1993 08:00 WASTE-TREATMENT-WC-2 -> AUDIT-THYSELF
Jan  3, 1993 08:00 SHIPPING-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 LATHE-LOCATION -> AUDIT-THYSELF
Jan  3, 1993 08:00 KIM-WIPES    -> AUDIT-THYSELF
Jan  3, 1993 08:00 COMPACTING-WC-2 -> AUDIT-THYSELF
Jan  3, 1993 08:00 SOLVENT      -> AUDIT-THYSELF
Jan  3, 1993 08:00 LATHE-WC-2   -> AUDIT-THYSELF
Jan  3, 1993 08:00 RECEIVING-LOCATION -> AUDIT-THYSELF
Jan  4, 1993 07:00 REPORT-GENERATOR -> TRIGGER NIL NIL
?
```

APPENDIX B

In-Class Problem Set

Constructing A Simulation Model

MODEL SCENARIO

Waste Management and Minimization

AT A STEEL PLANT

Student Problem Set

The class will build one complete simulation model during class as an example of how to use PWA. You will begin the class problem set with a simple scenario which we will construct and review together in class. Following this, you will construct and run scenarios for four additional problems. Successive problems are designed to be progressively more complex to allow you to learn the major facilities of PWA. The skills mastered in each problem will prepare you to solve more complex problems and will build upon the skills you develop. By the completion of this problem set, you will have mastered the basic functions and tools of PWA which will give you the ability to design and run simulation models of real-world functions and processes in your work place.

The processing system you will model is a small steel plant which has its own built-in waste treatment and waste storage facilities. The steel plant produces various steel alloys and parts, and performs lathing, casting and compacting.

The Functional Specification for this system is:

The plant produces two metal types, lathed steel parts and compacted steel parts from raw steel parts. Steel is purchased from a vendor in 100-pound parts. On January 1, 1993, there are 3 raw parts in receiving which have been inspected and are ready for processing in the plant.The steel plant is comprised of the following locations:

The Receiving location	Receives and stores in-coming raw steel parts
The Lathe location	Lathes the raw steel parts
The Shipping location	Stores and ships processed steel parts
The Waste treatment location	Stores and treats the chemical and non-chemical resources and by-products used in the processing of the steel parts

The Waste storage location	Stores the output (treated waste) from the Waste Treatment location.

The processes use both constrained and unconstrained resources. Steel parts move as both individual parts and as batches from bulk containers. .

PROBLEM 1

Using the PWA, model:

1) The movement and processing of parts.

1. Move three raw parts which are stored in the Receiving Location to the Lathe Location. Each raw part contains 100 units of steel.

2. Transform the raw parts into lathed parts at the Lathe Location. This process takes 8 hours.

3. Move the lathed parts to the Shipping Location.

Learning Objectives:

Learn how to move parts from one location to another

Learn how to process parts at a location using one workcenter

PROBLEM 2

Using Problem 1, model:

(a) The movement and processing of parts, and

(b) The movement of material into a bulk container in another location.

The Lathe Location now requires five units of machine oil in order to lathe the raw parts.All of the machine oil and one percent of the steel from the lathing operation goes into the waste tteatment vat, a bulk container in the waste treatment location. Ninety-nine percent of the steel goes with the lathed parts to the Shipping Location.

:

Learning Objectives:

Learn how to apply resources and direct output to a bulk container in another location.

PROBLEM 3

Use Problem 2, Model:

(a) The movement and processing of parts;

(b) The movement of material into a bulk container at another location;

(c) The processing of material from a bulk container by two identical workcenters in the same location;

(d) The movement of material into a bulk container in the same location; and

(e) The auditing and reporting of parts, resources, locations and workcenters.

1) Add kim-wipes and solvent as constrained resources. Put 1000 kim-wipes and 250 units of solvent into stock at the beginning of the simulation.

2) Add two identical workcenters in the waste treatment location. Each workcenter will process material from the waste treatment vat. The batch size is two units, based on the amount of steel in the waste treatment vat. The minimum batch size is two units. The process time is two hours. Each workcenter will require 5 kim-wipes and 24 units of solvent.

3) The output of each workcenter in the waste treatment location is:

(a) Send a new bag of waste (defined as a part) to the wastestorage location. This bag of waste contains all the kim-wipes, all the solvent, and all the machine oil.

(b) Send 100% of the steel to the cleaned steel vat which is located in the Waste Treament location.

4) Add auditors for all parts, resources, locations and workcenters. Set the audit period to daily.

5) Add all report-generating capabilities.

6) Change the end of the simulation to April 4, 1993, at 8:00 AM.

Learning Objectives:

Learn how to process different parts in a location that contains two identical workcenters.

Learn how to use auditors to monitor the collection of data for subsequent report generation.

Learn how to generate various kinds of reports.

PROBLEM 4

Editing the scenario you wrote in Problem 3, model:

(a) The movement and processing of parts;

(b) The movement of material into a bulk container in another location;

(c) The processing of material from a bulk container by two identical workcenters in the same location;

(d) The movement of material into a bulk container in the same location;

(e) The auditing and reporting of parts, resources, locations and workcenters; and

(f) A location with 3 workcenters, two of which draw material from one bulk container (see (c)), and another workcenter which draws material from a second bulk container.

1) Add a compacting workcenter in the waste treatment location. This workcenter will process material from the cleaned steel vat. The batch size is 1 unit, based on the total amount of material in the cleaned steel vat. The minimum batch size is 1 unit. The process time is 2 hours.

2) The output of the compacting workcenter in the waste treatment location is a new compacted steel part which contains all of the steel. Move this new part to the Lathe Location in 2 hours.

Learning Objectives:

Learn how to model the processing of two workcenters that perform different tasks and produce different outputs.

Problem 5

Using problem 4, model:

(a) The movement and processing of parts;

(b) The movement of material into a bulk container in another location;

(c) The processing of material from a bulk container by two identical workcenters in the same location;

(d) The movement of material into a bulk container in the same location;

(e) The auditing and reporting of parts, resources, locations and workcenters;

(f) A location with 3 workcenters, two of which draw material from one bulk container (see (c)), and another workcenter which draws material from a second bulk container; and

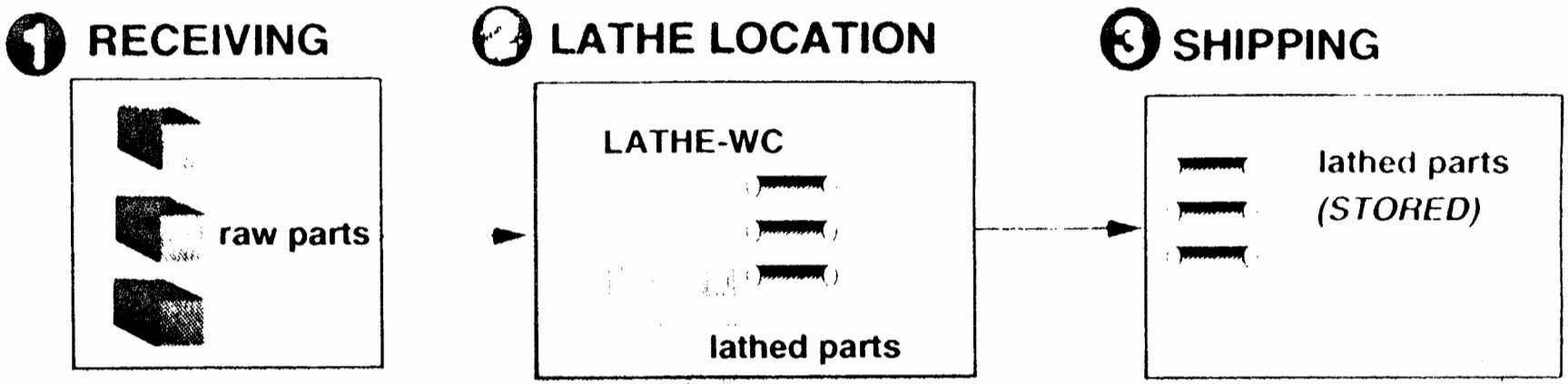
(g) A workcenter which can perform two different processes.

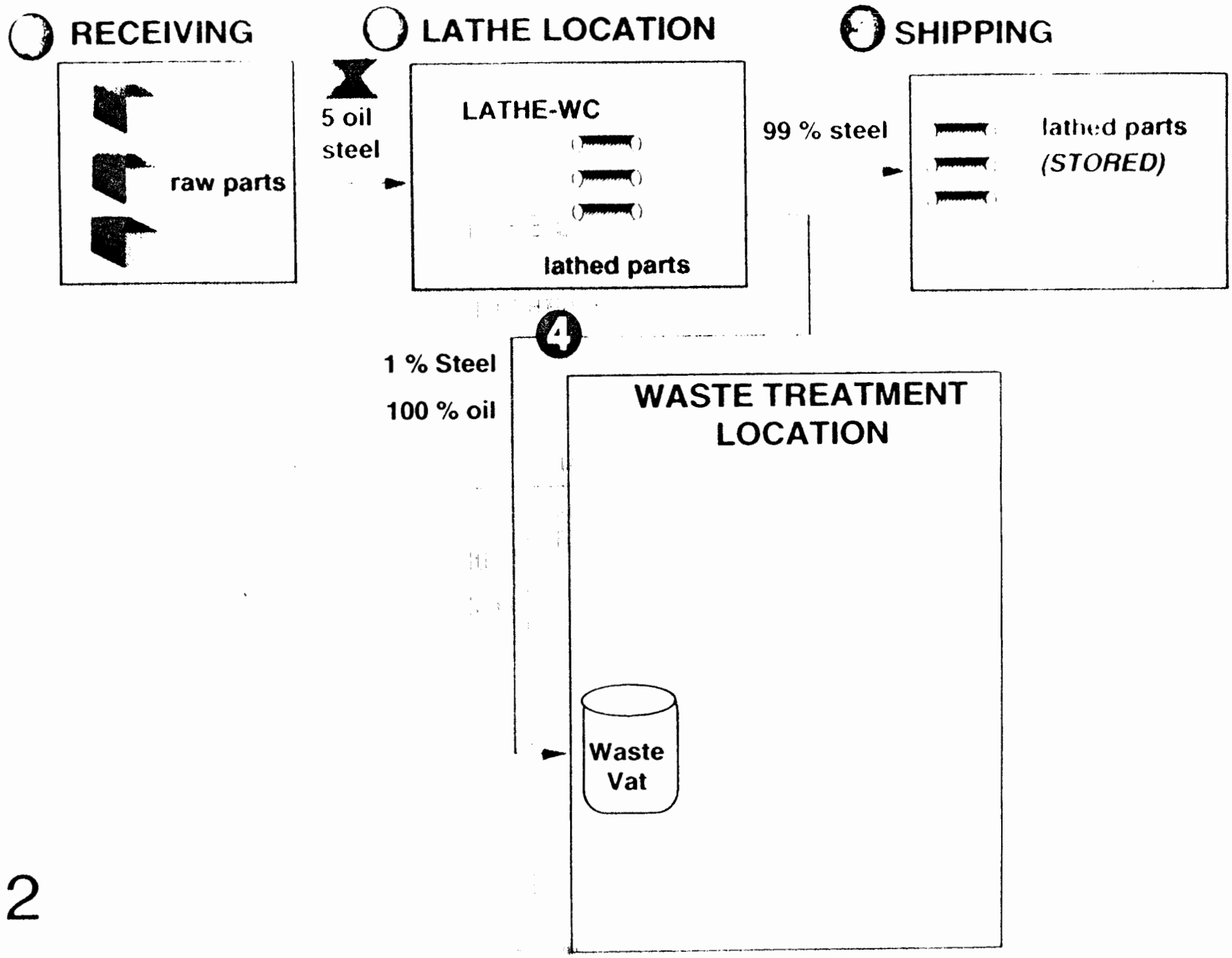
1) Add a new process step description to the lathe workcenter in the lathe location to transform compacted steel parts into lathed compacted parts: Lathing the compacted steel parts requires 5 units of machine oil. The process takes 30 minutes. All of the machine oil and 1% of the steel from compacting the steel parts goes into the waste treatment vat in the waste treatment location. 99% of the steel goes with the lathed compacted parts to the shipping location.

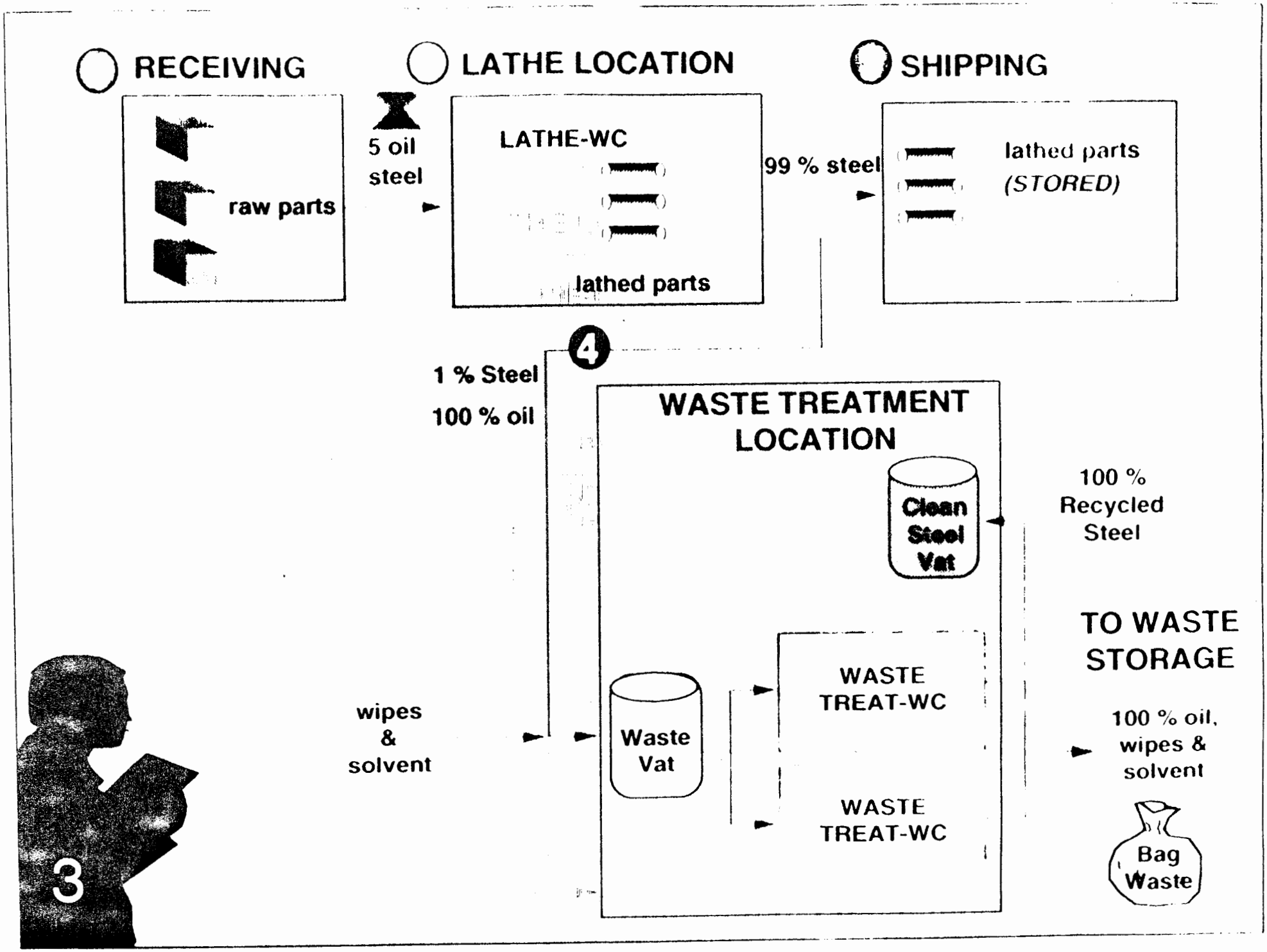
Learning Objectives:

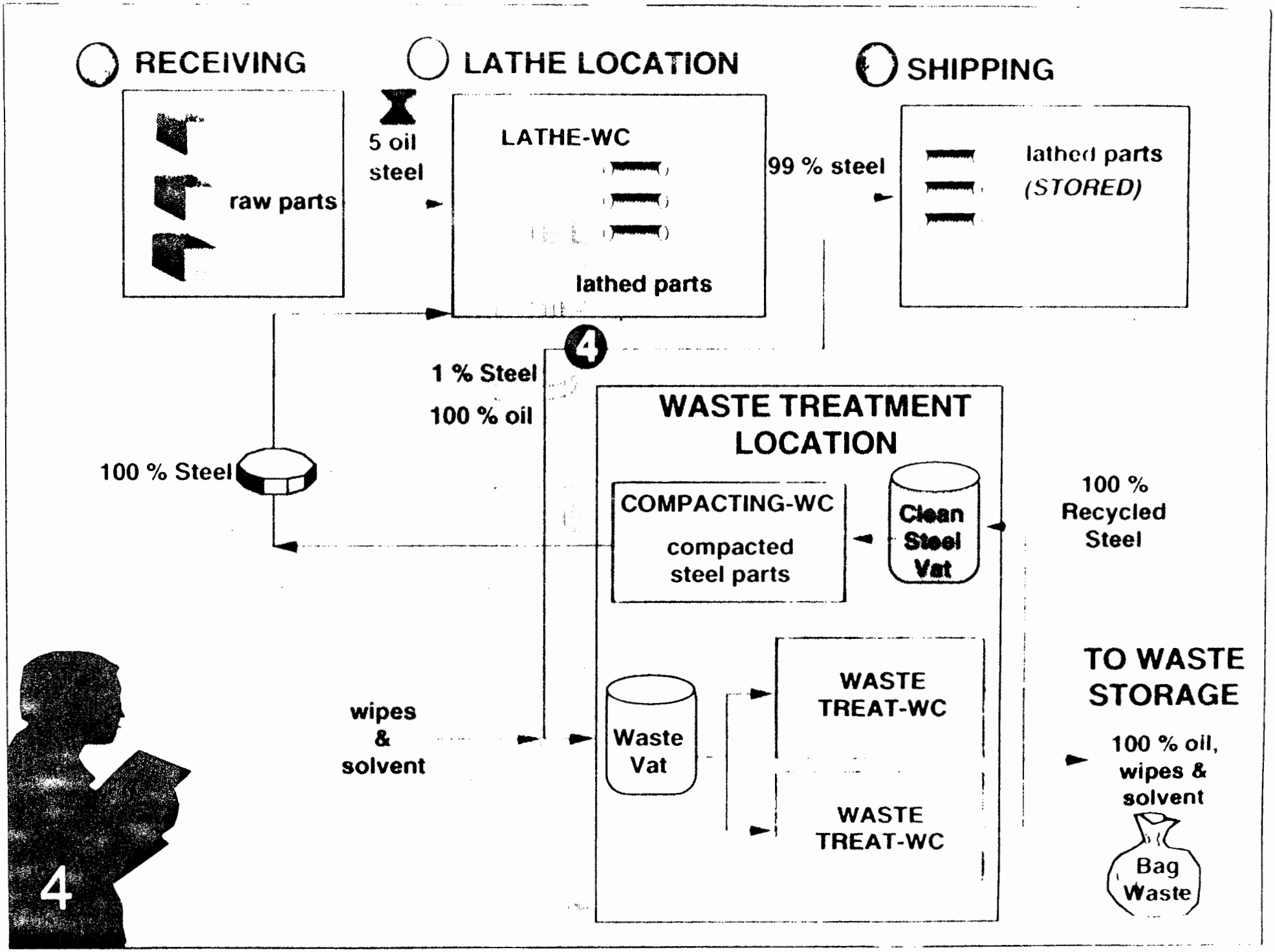
Learn how to use PWA to model the processing so that a workcenter can perform two different tasks.

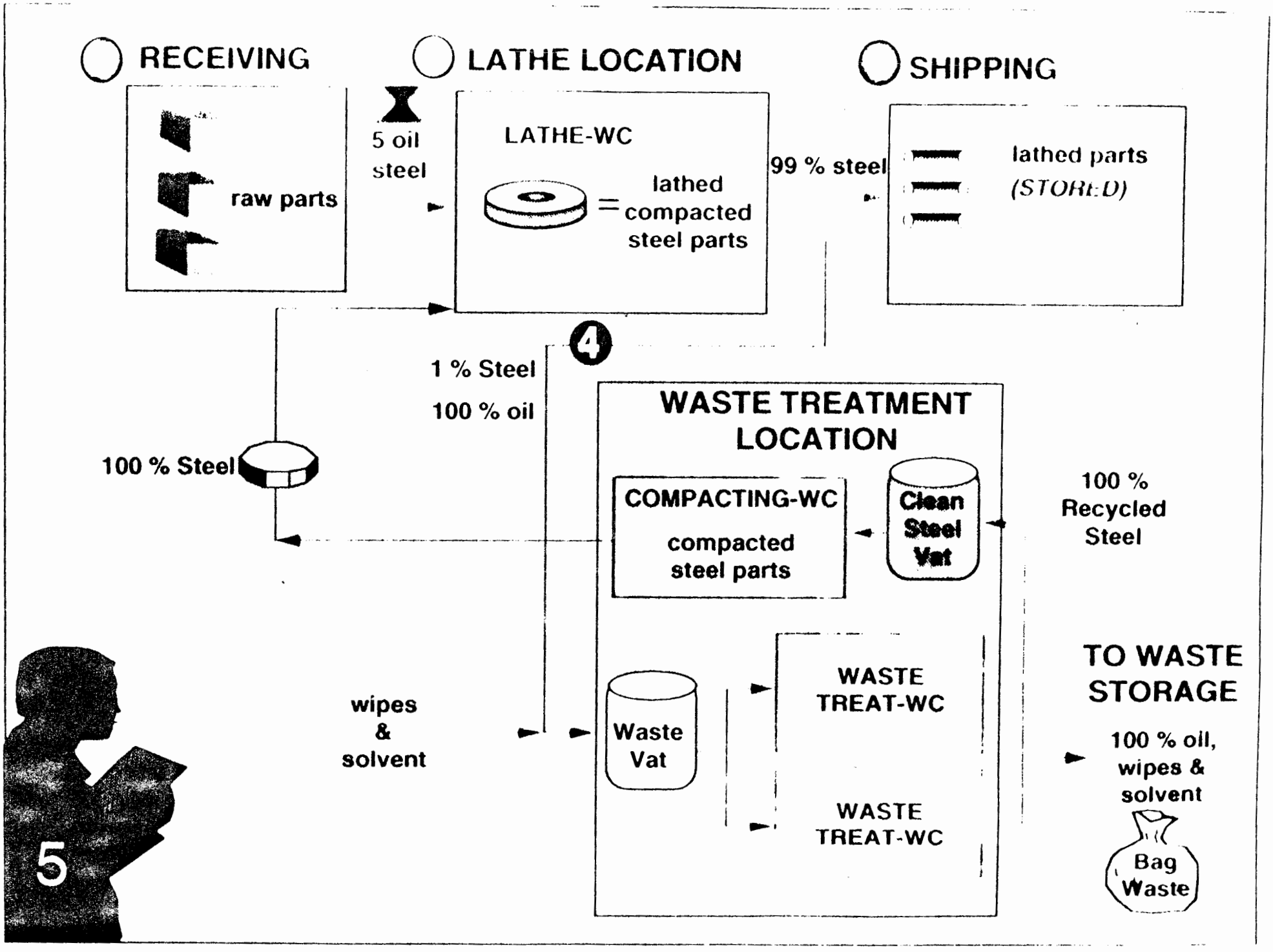
SOLUTIONS TO PROBLEMS


1 RECEIVING
raw parts
2 LATHE LOCATION
LATHE-WC
lathed parts
3 SHIPPING
lathed parts
(STORED)


RECEIVING
raw parts
5 oil
steel
LATHE LOCATION
LATHE-WC
lathed parts
99 % steel
SHIPPING
lathed parts
(STORED)
4
1 % Steel
100 % oil
WASTE TREATMENT
LOCATION
Waste
Vat


RECEIVING
LATHE LOCATION
SHIPPING
raw parts
5 oil
steel
LATHE-WC
lathed parts
99 % steel
lathed parts
(STORED)
4
1 % Steel
100 % oil
WASTE TREATMENT
LOCATION
Clean
Steel
Vat
100 %
Recycled
Steel
TO WASTE
STORAGE
wipes
&
solvent
Waste
Vat
WASTE
TREAT-WC
WASTE
TREAT-WC
100 % oil,
wipes &
solvent
Bag
Waste
3


RECEIVING
LATHE LOCATION
SHIPPING
raw parts
5 oil
steel
LATHE-WC
lathed parts
99 % steel
lathed parts
(STORED)
4
1 % Steel
100 % oil
100 % Steel
WASTE TREATMENT
LOCATION
COMPACTING-WC
compacted
steel parts
Clean
Steel
Vat
100 %
Recycled
Steel
wipes
&
solvent
Waste
Vat
WASTE
TREAT-WC
WASTE
TREAT-WC
TO WASTE
STORAGE
100 % oil,
wipes &
solvent
Bag
Waste
4


RECEIVING
LATHE LOCATION
SHIPPING
raw parts
5 oil
steel
LATHE-WC
= lathed compacted steel parts
99 % steel
lathed parts
(STORED)
4
1 % Steel
100 % oil
100 % Steel
WASTE TREATMENT LOCATION
COMPACTING-WC
compacted steel parts
Clean Steel Vat
100 % Recycled Steel
wipes & solvent
Waste Vat
WASTE TREAT-WC
WASTE TREAT-WC
TO WASTE STORAGE
100 % oil, wipes & solvent
Bag Waste
5

PWA CHALLENGE PROBLEM

Supplementary Student Problem

EM_PROBLEM 1

USING PWA, MODEL THE MOVEMENT AND PROCESSING OF THREE RAW PARTS:

1) MOVE THE PARTS FROM THE RECEIVING LOCATION TO THE LATHE LOCATION
2) PROCESS THE PARTS AT THE LATHE LOCATION.
3) MOVE THE LATHED PARTS TO THE SHIPPING LOCATION

Learning Objectives:

Learn how to move parts from one location(s) to another.
Learn how to process parts at a location using one workcenter.
Learn how to apply unconstrained resources to a process.

EXPLANATION OF SYNTAX

SET THE CLOCK - HOW LONG DO I WANT TO RUN THIS SIMULATION?

:CLOCK	(0.0 0 8 1 1 1993)	Start the simulation on January 1, 1993 at 8:00 am
:LIMIT	(0.0 0 8 8 1 1993)	End the simulation on January 8, 1993, at 8:00 am

EXPLANATION OF SYNTAX

DEFINE THE PARTS TO BE PROCESSED:

:CLASS our-parts **:PARENTS** (parts)	Define a type of part and define the parents
:SUBCLASS raw-parts	Define a type of our-parts
PART-MATERIALS ((steel 100))	Raw-parts each contain 100 units of steel
WORKORDER lathe-raw-parts-wo	Raw-parts each move and are processed as described by the lathe-raw-parts-wo
:END-SUBCLASS raw-parts	End the definition of raw parts
:SUBCLASS lathed-parts	Define another type of our-parts
:END-SUBCLASS lathed-parts	End the subclass

EXPLANATION OF SYNTAX

DEFINE THE LOCATIONS BETWEEN WHICH PARTS ARE MOVED:

`:CLASS my-locations` `:PARENTS (locations)`	Define a type of locations
`:instance receiving-location`	Create one location called receiving
`stored-parts ((raw-parts 3 done))`	Receiving will initially contain three raw parts. The status "done" indicates that the raw parts can be moved to another location
`:at current-simulation-clock` `decide-operation`	Post an event on the list of things to be done which will allow receiving to determine if it should move or process some material.
`:instance lathe-location`	Create a type of locations called the lathe-location
`attached-workcenters ((lathe-wc))`	The lathe location will contain one piece of equipment of the type lathe-wc

	EXPLANATION OF SYNTAX
:instance shipping-location	Create a type of locations called the shipping-location

DEFINE THE WORKORDERS WHICH TELL THE LOCATIONS WHERE TO MOVE THE PARTS AND HOW TO PROCESS THE PARTS

:CLASS my-workorders **:PARENTS** (workorders)	Define a type of workorders
:SUBCLASS lathe-raw-parts-wo	Define a type of my-workorders
workorder-steps	This workorder describes two steps
((**transfer** lathe-location (0 15))	The first step will move the part to which the workorder is attached to the lathe-location in 15 minutes
(lathe-a-part lathe-location (0 0 8)))	The second step allows the part to be processed by any piece of equipment at the lathe-location that knows how to lathe-a-raw-part. This process takes 8 hours
:END-SUBCLASS lathe-raw-parts-wo	End the subclass

EXPLANATION OF SYNTAX

:SUBCLASS move-to-shipping-wo	Define another type of my-workorders
workorder-steps ((**transfer** shipping-location (0 15 0)))	This step transfers the lathed parts to which the workorder is attached to the shipping-location in 15 minutes
:END-SUBCLASS move-to-shipping-wo	End the definition of move-to-shipping-wo

DEFINE THE PIECES OF EQUIPMENT THAT WILL PROCESS THE MATERIAL
ALL WORKCENTERS ARE MASS BALANCED (AMOUNT OF MATERIAL IN THE INPUTS = AMOUNT OF MATERIAL IN THE OUTPUTS)

:CLASS my-mass-balance-workcenters	Define a type of mass-balance-workcenters
:PARENTS (mass-balance-workcenters) **:SUBCLASS** lathe-wc	Define the workcenter that will perform the process of lathing the parts

EXPLANATION OF SYNTAX

process-step-descriptions	Begin defining the steps for processing the part

DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

((FREE	This workcenter must be in a free state to begin processing

DESCRIBE THE INPUTS:

((*temp-part* lathe-a-part))	This workcenter can work on a stored-part that has *lathe-a-part* as its current workorder entry. *Temp-part* is a local name assigned by the user. It is used only within this process step, allowing reference to this input by later sections.
()	Place holder for future expansion of the model

EXPLANATION OF SYNTAX

MATERIAL RESOURCES:

()	This workcenter requires no material resources

REUSABLE RESOURCES:

()	This workcenter requires no reusable resources

DESCRIBE THE OUTPUTS OF THE WORKCENTER:

(FREE	This workcenter will be left in a free state at the end of this processing step
`steel`	The workcenter will distribute any materials, not specifically described in its output mass balance, relative to the amount of steel in each output. Could also use nil () defaulting rule.

EXPLANATION OF SYNTAX

`(:new lathed-parts`	Create a new instance of lathed-parts
`move-to-shipping-wo`	The name of the workorder to be attached to lathed-parts
`()`	The part-attributes of lathed-parts. () or nil indicates no attributes are specified
`(steel 1.0)))))`	Put all the steel from the input into the new instance of lathed-parts
`:END-SUBCLASS lathe-wc`	End the definition of lathe-wc

EM_PROBLEM 2

USING PWA, MODEL THE MOVEMENT AND PROCESSING OF THREE RAW PARTS:

THE LATHE LOCATION NOW REQUIRES 5 UNITS OF MACHINE OIL IN ORDER TO LATHE THE RAW PARTS. THE MACHINE OIL IS A TOXIC RCRA LISTED MATERIAL THAT CARRIES EPA HAZARD WASTE NUMBERS F003 AND F004. MACHINE OIL IS USED BY THE PINT.

Learning Objectives:
Learn how to direct output to a bulk container in another location
Learn how to apply an unconstrained resource

;;SET THE CLOCK - HOW LONG DO I WANT TO RUN THIS SIMULATION?

	EXPLANATION OF SYNTAX
:CLOCK (0.0 0 8 1 1 1993)	;;Start the simulation on January 1, 1993 at 8:00 am
:LIMIT (0.0 0 8 8 1 1993)	;;End the simulation on January 4, 1993, at 8:00 am

EXPLANATION OF SYNTAX

;;DEFINE THE PARTS TO BE PROCESSED:

```
:CLASS      our-parts                          ;;Define a type of parts and define the parents
:PARENTS (parts)

 :SUBCLASS    raw-parts                         ;;Define a type of our-parts

      PART-MATERIALS   ((steel 100))            ;;Raw-parts each contain 100 units of steel

      WORKORDER    lathe-raw-parts-wo           ;;Raw-parts each move and are processed as
                                                ;;described by the lathe-raw-parts-wo

 :END-SUBCLASS    raw-parts                     ;;End the definition of raw parts

  :SUBCLASS    lathed-parts                     ;;Define a type of our-parts

  :END-SUBCLASS    lathed-parts                 ;;End the subclass
```

EXPLANATION OF SYNTAX

;;DEFINE AVAILABLE RESOURCES WITH UNLIMITED AMOUNTS WHICH CAN BE ADDED ;;TO PROCESSES:

```
:CLASS   my-factory-resources          ;;Define a type of em-factory-resources
:PARENTS  (resources PWA-materials)

      :instance machine-oil            ;;Create a type of unconstrained resources
          epa-hw (F003 F004)           ;;Assign the applicable EPA numbers
          material-class toxic         ;;The material class is of the type toxic
          rcra-listed-material T       ;;The attribute rcra-listed-material is true
          units pints                  ;;The oil comes in pints
```

;;DEFINE THE LOCATIONS BETWEEN WHICH PARTS ARE MOVED:

```
:CLASS   my-locations                  ;;Define a type of locations
:PARENTS  (locations)

     :instance  receiving-location     ;;Create a type of locations called receiving-location

     stored-parts ((raw-parts 3 done))  ;;Receiving-location will initially contain three raw
                                        ;;parts. The status "done" indicates that the raw
                                        ;;parts can be moved to another location
```

EXPLANATION OF SYNTAX

```
:at current-simulation-clock                ;;Post an event on the list of things to be
 decide-operation                           ;;done which will allow the receiving-location to
                                            ;;determine if it should move or process

  :instance    lathe-location                ;;Create a type of locations called the lathe-
                                            ;;location

     attached-workcenters ((lathe-wc))       ;;The lathe-location will contain one piece of
                                            ;;equipment of the type lathe-wc

   :instance    shipping-location            ;;Create a type of locations called the shipping-
                                            ;;location

   :instance   waste-treatment-location     ;;Create a type of locations called the waste-
                                            ;;treatment-location

     stored-bulk ((waste-treatment-vat))    ;;The waste-treatment-location contains one

;;DEFINE THE WORKORDERS WHICH TELL THE LOCATIONS WHERE TO MOVE THE
;;PARTS AND HOW TO PROCESS THE PARTS

:CLASS    my-workorders                     ;;Define a type of workorders
:PARENTS   (workorders)
```

EXPLANATION OF SYNTAX

```
:SUBCLASS    lathe-raw-parts-wo          ;;Define a type of my-workorders

  workorder-steps                        ;;This workorder describes two steps.

   ((transfer  lathe-location (0 15))    ;;The first step will move the part to
                                         ;;which the workorder is attached to the lathe
                                         ;;location in 15 minutes

   (lathe-a-part  lathe-location
        (0   0 8)))                      ;;The second step allows the part to be processed by
                                         ;;any piece of equipment at the lathe location that
                                         ;;knows how to lathe-a-raw-part.  This process
                                         ;;takes 8 hours.

  :END-SUBCLASS   lathe-raw-parts-wo     ;;End the subclass

   :SUBCLASS    move-to-shipping-wo       ;;Define another type of my-workorders

    workorder-steps
    ((transfer  shipping-location        ;;This step transfers the lathed parts to which
        (0 15 0)))                       ;;the workorder is attached to the shipping
                                         ;;location in 15 minutes
```

EXPLANATION OF SYNTAX

```
    :END-SUBCLASS  move-to-shipping-wo      ;;End the definition of move-to-shipping-wo

;;DEFINE THE PIECES OF EQUIPMENT THAT WILL PROCESS THE MATERIAL.
;;ALL WORKCENTERS ARE MASS BALANCED (AMOUNT OF MATERIAL IN THE INPUTS =
;;AMOUNT OF MATERIAL IN THE OUTPUTS)

:CLASS    my-mass-balance-workcenters   ;;Define a type of mass-balance-workcenters
:PARENTS   (mass-balance-workcenters)

    :SUBCLASS   lathe-wc                ;;Define the workcenter that will perform the
                                        ;;process of lathing the parts

process-step-descriptions               ;;Begin defining the steps for processing
                                        ;;the part

;;BEGIN BY DESCRIBING THE INITIAL STATE OF THE EQUIPMENT:

((FREE                                  ;;This workcenter must be in a free state
                                        ;;to begin processing
```

EXPLANATION OF SYNTAX

;;DESCRIBE THE INPUTS:

```
((temp-part   lathe-a-part))
```

;;This workcenter can work on a stored-part that
;;has lathe-a-part as its current workorder entry.
;;*Temp-part* is a local name assigned by the user.
;;It is used only within this process step, allowing
;;reference to this input by later sections.

()

;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

```
((machine-oil 5))
```

;;This workcenter requires 5 units of
;;machine oil from the stockroom
;;(resources). This machine oil will be
;;incorporated into the process output

;;REUSABLE RESOURCES:

()

;;This workcenter requires no reusable resources

;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:

(FREE

;;This workcenter will be left in a free state at
;;the end of this processing step

EXPLANATION OF SYNTAX

`steel`	;;The workcenter will distribute any materials, ;;not specifically described in its output mass ;;balance, relative to the amount of steel in each ;;output
`(:new lathed-parts`	;;Create a new instance of lathed-parts
`move-to-shipping-wo`	;;The name of the workorder to be attached ;;to lathed-parts
`()`	;;The part-attributes of lathed-parts. ()
`(steel .99))`	;;Put 99% of the steel into lathed-parts
(:feed waste-treatment-location	;;Send to a bulk container in the waste-treatment ;;-location
waste-treatment-vat	;;Assign a name to the bulk container
(steel .01) (machine-oil 1.0)))))	;;Place 1% of the steel and 100% of the ;;machine-oil in the bulk container
:END-SUBCLASS `lathe-wc`	;;End the definition of lathe-wc

EM_PROBLEM 3

USE PWA TO MODEL THE PROCESSING OF THE PARTS UNDER THE FOLLOWING SCENARIO:

1) ADD TWO IDENTICAL WORKCENTERS (PERFORM THE SAME PROCESS) IN THE WASTE TREATMENT LOCATION. EACH WORKCENTER WILL PROCESS MATERIAL FROM THE WASTE TREATMENT VAT. THE BATCH SIZE IS TWO UNITS AND IS BASED ON THE AMOUNT OF STEEL IN THE WASTE TREATMENT VAT. THE MINIMUM BATCH SIZE IS TWO UNITS. THE PROCESS TIME IS TWO HOURS. EACH WORKCENTER WILL REQUIRE FIVE KIM-WIPES AND 24 UNITS OF SOLVENT.

2) ADD KIM-WIPES AND SOLVENT AS CONSTRAINED RESOURCES. PUT 1000 KIM-WIPES AND 250 UNITS OF SOLVENT INTO STOCK AT THE BEGINNING OF THE SIMULATION.

3) THE OUTPUT OF EACH WORKCENTER IN THE WASTE TREATMENT LOCATION IS:
(A) SEND A NEW BAG OF WASTE (DEFINED AS A PART) TO THE WASTE STORAGE LOCATION. THIS BAG OF WASTE CONTAINS ALL THE KIM-WIPES, ALL THE SOLVENT, AND ALL THE MACHINE OIL.
(B) SEND 100% OF THE STEEL TO THE CLEANED STEEL VAT WHICH IS LOCATED IN THE WASTE TREATMENT LOCATION.

4) ADD AUDITORS FOR ALL PARTS, RESOURCES, LOCATIONS AND WORKCENTERS. SET THE AUDIT PERIOD TO DAILY.

5) ADD ALL REPORT-GENERATING CAPABILITIES.

Learning Objectives:
Learn how to apply unconstrained resources to a process.
Learn how to direct output to a bulk container in another location
Learn how to process different parts in a location that contains two identical workcenters.
Learn how to use auditors to monitor the collection of data for subsequent report generation

EXPLANATION OF SYNTAX

;;SET THE CLOCK:

```
:CLOCK   (0.0 0 8 1 1 1993)        ;;Start the simulation on January 1, 1993 at 8:00 am
:LIMIT   (0.0 0 8 4 1 1993)        ;;End the simulation on January 4, 1993, at 8:00 am

:CLASS    my-foremen               ;;Define a type of foremen that will both audit
:PARENTS  (reporter-foremen        ;;and generate reports when triggered
            auditing-foremen)

     :instance  report-generator   ;;report-generator will generate reports
                                   ;;according to the following list of report
                                   ;;types and report file names:
```

EXPLANATION OF SYNTAX

```
processing-cumulative   "CUMULATIVE-REPORT-Problem-3"   ;;a report of cumulative
                                                        ;;processing information for
                                                        ;;each location

processing-incremental  "INCREMENTAL-REPORT-Problem-3"  ;;a report of incremental
                                                        ;;processing information for
                                                        ;;each location

feed                    "FEED-REPORT-Problem-3"         ;;a report of amounts and sources
                                                        ;;of all material for each location

materials               "MATERIALS-REPORT-Problem-3"    ;;a report of cumulative material
                                                        ;;holdings for each location

material-handling       "MATERIAL-HANDLING-REPORT-Problem-3"
                                                        ;;a report of material processing
                                                        ;;and utilization for each location

part-handling           "PART-HANDLING-REPORT-Problem-3"
                                                        ;;an inventory of parts for each
                                                        ;;location
```

EXPLANATION OF SYNTAX

:at (0 0 7 4 1 1993) **(trigger nil nil)**	;;Post an event on the list of things to be done which ;;will allow the report-generator to perform a final ;;audit and generate reports
;;DEFINE THE PARTS:	
:CLASS my-parts	;;Define a type of parts and define the parents
:PARENTS (parts audited-parts) **audit-period :daily**	;;Instances of my-parts are audited daily
:SUBCLASS raw-parts	;;Define a type of my-parts
PART-MATERIALS ((steel 100))	;;Raw-parts each contain 100 units of steel
WORKORDER lathe-raw-parts-wo	;;Raw-parts each move and are processed as ;;described by the lathe-raw-parts-wo
:END-SUBCLASS raw-parts	;;End the definition of raw-parts
:SUBCLASS lathed-parts	;;Define a type of my-parts
:END-SUBCLASS lathed-parts	;;End the subclass
:SUBCLASS bag-of-waste	;;Define a type of my-parts
:END-SUBCLASS bag-of-waste	;;End the subclass

EXPLANATION OF SYNTAX

;;DEFINE AVAILABLE RESOURCES WITH UNLIMITED AMOUNTS WHICH CAN BE ADDED ;;TO PROCESSES:

```
:CLASS  my-factory-resources                ;;Define a type of audited unconstrained resources
:PARENTS  (resources audited-resources
              PWA-materials)

    audit-period :daily                     ;;Instances of my-factory-resources are audited on
                                            ;;a daily basis

:instance machine-oil                       ;;Create a type of unconstrained resources
          epa-hw (F003 F004)                ;;Assign the applicable EPA numbers
          material-class toxic              ;;The material class is of the type toxic
          rcra-listed-material T            ;;The attribute rcra-listed-material is true
          units pints                       ;;The oil comes in pints
```

;;DEFINE AVAILABLE RESOURCES WITH LIMITED AMOUNTS:

```
:CLASS  my-factory-constrained-resources;;Define a type of audited resources whose available
                                         ;;amounts are limited
:PARENTS  (resources-with-amounts
               audited-resources)        ;;Create the parent class

audit-period :daily                      ;;Instances of my-factory-resources are audited
                                         ;;daily
```

	EXPLANATION OF SYNTAX
:instance kim-wipes	;;Create a type of constrained resources called kim- ;;wipes
available-resource 1000	;;There are initially 1000 units of kim-wipes ;;available
:instance solvent **available-resource 250**	;;Create a type of constrained resources called ;;250 units are available

;;DEFINE THE LOCATIONS BETWEEN WHICH PARTS ARE MOVED:

:CLASS my-locations **:PARENTS** (locations audited-locations)	;;Define a type of audited locations
audit-period :daily	;;Instances of my-locations are audited daily
:instance receiving-location	;;Create a type of locations called receiving-location
stored-parts ((raw-parts 3 done))	;;Receiving-location will initially contain three ;;raw-parts. The status "done" indicates that the ;;raw-parts can be moved to another location
:at current-simulation-clock **decide-operation**	;;Post an event on the list of things to be ;;done which will allow the receiving-location to ;;determine if it should move or process ;;some material.

EXPLANATION OF SYNTAX

```
:instance    lathe-location                  ;;Create another type of locations called the lathe
                                             ;;location

    attached-workcenters ((lathe-wc))         ;;The lathe-location will contain one piece of
                                              ;;equipment of the type lathe-wc


  :instance    shipping-location              ;;Create another type of locations called the shipping
                                              ;;-location

  :instance   waste-treatment-location        ;;Create another type of locations called the waste-
                                              ;;treatment-location

    stored-bulk ((waste-treatment-vat)        ;;The waste-treatment-location contains two
      (cleaned-steel-vat))                    ;;bulk containers,  the waste-treatment-vat
                                              ;;and the cleaned-steel-vat

    attached-workcenters
      ((waste-treatment-wc)
        (waste-treatment-wc))                 ;;The waste-treatment location contains two
                                              ;;identical pieces of equipment of the type waste-
                                              ;;treatment-wc

:instance   waste-storage-location            ;;Create another type of locations called the waste-
                                              ;;storage-location

;;DEFINE THE WORKORDERS WHICH TELL THE LOCATIONS WHERE TO MOVE AND PROCESS
;;THE PARTS:
```

EXPLANATION OF SYNTAX

```
:CLASS    my-workorders                    ;;Define a type of workorder

:PARENTS  (workorders)                     ;;Define the parent class

   :SUBCLASS    lathe-raw-parts-wo         ;;Define a type of my-workorders

   workorder-steps                         ;;This workorder describes two steps

     ((transfer  lathe-location (0 15))    ;;The first step will move the part to
                                           ;;which the workorder is attached to the lathe
                                           ;;location in 15 minutes

     (lathe-a-raw-part  lathe-location
          (0  0 8)))                       ;;The second step allows the part to which the
                                           ;;workorder is attached to be processed by any
                                           ;;piece of equipment at the lathe location that
                                           ;;knows how to lathe-a-raw-part

 :END-SUBCLASS   lathe-raw-parts-wo        ;;End the subclass

   :SUBCLASS    move-to-shipping-wo         ;;Define another type of my-workorders

   workorder-steps
     ((transfer    shipping-location       ;;This step moves the part to which
         (0 15 0)))                        ;;the workorder is attached to the shipping-
                                           ;;location in 15 minutes
```

EXPLANATION OF SYNTAX

```
:END-SUBCLASS  move-to-shipping-wo          ;;End the definition of move-to-shipping-wo

   :SUBCLASS   move-to-waste-storage-wo     ;;Define another type of my-workorders

   workorder-steps
      ((transfer waste-storage-location     ;;This step moves the part  to which
          (0 0 2)))                         ;;the workorder is attached to the waste-storage-
                                            ;;location in 2 hours

   :END-SUBCLASS move-to-waste-storage-wo
                                            ;;End the definition of move-to-waste-storage-wo

;;DEFINE THE PIECES OF EQUIPMENT THAT WILL PROCESS THE MATERIAL.
;;ALL WORKCENTERS ARE MASS BALANCED (AMOUNT OF MATERIAL IN THE INPUTS =
;;AMOUNT OF MATERIAL IN THE OUTPUTS):


:CLASS    my-mass-balance-workcenters       ;;Define a type of mass balance workcenters
:PARENTS    (mass-balance-workcenters
          audited-workcenters)

   audit-period :daily                      ;;Instances of my-mass-balance-workcenters
                                            ;;are audited daily

    :SUBCLASS   lathe-wc                    ;;Define the workcenter that will  lathe the parts

  process-step-descriptions                 ;;Begin defining the steps for processing
                                            ;;the part
```

EXPLANATION OF SYNTAX

;;DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

((FREE	;;This workcenter must be in a free state ;;to begin processing

;;DESCRIBE THE INPUTS:

((*temp-part* lathe-a-raw-part))	;;This workcenter can work on a stored-part that ;;has lathe-a-raw-part as its current workorder ;;entry. ;;*Temp-part* is a local name assigned by the user ;;used only within this process step, allowing ;;reference to this input by later sections.
()	;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

((machine-oil 5))	;;This workcenter requires 5 units of ;;machine oil from the stockroom. The machine oil ;;will be incorporated into the process output

;;REUSABLE RESOURCES:

()	;;This workcenter requires no reusable resources

EXPLANATION OF SYNTAX

;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:

```
(FREE                                  ;;This workcenter will be left in a free state at
                                       ;;the end of this processing step

steel                                  ;;The workcenter will distribute any materials,
                                       ;;not specifically described in its output mass
                                       ;;balance, relative to the amount of steel in each
                                       ;;output

(:new lathed-parts                     ;;Create a new instance of lathed-parts

   move-to-shipping-wo                 ;;The name of the workorder to be attached to
                                       ;;lathed-parts

()                                     ;;The part-attributes of lathed-parts

(steel  .99))                          ;;Put 99% of the steel into lathed-parts

(:feed waste-treatment-location        ;;Send to a bulk-container in the waste-
                                       ;;treatment location

waste-treatment-vat                    ;;The name of the bulk container

(steel .01)                            ;;Put 1% of the steel into the bulk container

(machine-oil 1.0)))))                  ;;Put 100% of the steel into the bulk container
```

EXPLANATION OF SYNTAX

:END-SUBCLASS lathe-wc	;;End the definition of lathe-wc
:SUBCLASS waste-treatment-wc	;;Define the workcenter that will process ;;material from the waste-treatment-vat
process-step-descriptions	;;Begin defining the steps for processing ;;the part

;;DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

((FREE	;;This workcenter must be in a free state ;;to begin processing

;;DESCRIBE THE INPUTS:

((**:bulk** waste-treatment-vat	;;The name of the bulk container from which ;;to draw material
steel	;;The workcenter will draw material relative to ;;the amount of steel in the vat
2	;;The number of units of the basis material that ;;goes into each batch
nil	;;The amount of material that can be processed ;;in the time defined below -- specify nil or a ;;number

EXPLANATION OF SYNTAX

(0 0 2)	;;How long to process the amount
2))	;;The minimum batch size
()	;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

`((kim-wipes 5)` `(solvent 24))`	;;This workcenter requires 5 units of ;;kim-wipes and 24 units of solvent as ;;resources from the stockroom. Both resources will ;;be incorporated into the process output

;;REUSABLE RESOURCES:

()	;;This workcenter requires no reusable resources

;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:

(FREE	;;This workcenter will be left in a free state at ;;the end of this processing step
`steel`	;;The workcenter will distribute any materials, ;;not specifically described in its output mass ;;balance, relative to the amount of steel in each

EXPLANATION OF SYNTAX

```
(:new bag-of-waste                          ;;Create a new instance of bag-of-waste parts
  move-to-waste-storage-wo                  ;;The name of the workorder to be attached to
                                            ;;the bag-of-waste
()                                          ;;The part-attributes of the bag-of-waste
(kim-wipes 1.0)                             ;;Put all the kim-wipes,
(solvent 1.0)                               ;;all the solvent,
(machine-oil 1.0))                          ;;and all of the machine-oil into the
                                            ;;bag-of-waste
(:bulk cleaned-steel-vat                    ;;Send to a bulk container in this location
 (steel 1.0)))))                            ;;all of the steel,
 :END-SUBCLASS  waste-treatment-wc          ;;End the definition of waste-treatment-wc
```

EM_PROBLEM 4

```
:
;;USING PROBLEM 3, MODEL
;;    (a) THE MOVEMENT AND PROCESSING OF PARTS;
;;    (b) THE MOVEMENT OF MATERIAL INTO A BULK CONTAINER IN ANOTHER
;;        LOCATION;
;;    (c) THE PROCESSING OF MATERIAL FROM A BULK CONTAINER BY TWO
;;        IDENTICAL WORKCENTERS IN THE SAME LOCATION;
;;    (d) THE MOVEMENT OF MATERIAL INTO A BULK CONTAINER IN THE SAME
;;        LOCATION;
;;    (e) THE AUDITING AND REPORTING OF PARTS, RESOURCES, LOCATIONS AND
;;        WORKCENTERS; AND
;;    (f) A LOCATION WITH THREE WORKCENTERS, TWO OF WHICH DRAW MATERIAL
;;        FROM A SECOND BULK CONTAINER. (SEE (c)), AND ANOTHER WORKCENTER
;;        WHICH DRAWS MATERIAL FROM A SECOND BULK CONTAINER.

;;1. ADD A COMPACTING WORKCENTER IN THE WASTE TREATMENT LOCATION. THIS
;;   WORKCENTER WILL PROCESS MATERIAL FROM THE CLEANED STEEL VAT. THE
;;   BATCH SIZE IS ONE UNIT, BASED ON THE TOTAL AMOUNT OF MATERIAL IN THE
;;   CLEANED STEEL VAT. THE MINIMUM BATCH SIZE IS ONE UNIT. THE PROCESS
;;   TIME IS TWO HOURS.

;;2. THE OUTPUT OF THE COMPACTING WORKCENTER IN THE WASTE TREATMENT
;;   IS A NEW COMPACTED STEEL PART WHICH CONTAINS ALL OF THE STEEL. MOVE
;;   THIS NEW PART TO THE LATHE LOCATION IN TWO HOURS.
```

Learning Objectives:

Learn how to model the processing of two workcenters that perform different tasks and produce different outputs.

EXPLANATION OF SYNTAX

;;SET THE CLOCK:

```
:CLOCK   (0.0 0 8 1 1 1993)        ;;Start the simulation on January 1, 1993 at 8:00 am
:LIMIT   (0.0 0 8 4 1 1993)        ;;End the simulation on January 4, 1993, at 8:00 am

:CLASS    my-foremen               ;;Define a type of foremen that will audit
                                   ;;and generate reports when triggered

:PARENTS (reporter-foremen auditing-foremen)
  :instance  report-generator      ;;Report-generator will generate reports
                                   ;;according to the following list of report
                                   ;;types and report file names:


processing-cumulative              "CUMULATIVE-REPORT-Problem-4"
processing-incremental             "INCREMENTAL-REPORT-Problem-4"
feed                               "FEED-REPORT-Problem-4"
materials                          "MATERIALS-REPORT-Problem-4"
material-handling                  "MATERIAL-HANDLING-REPORT-Problem-4"
part-handling                      "PART-HANDLING-REPORT-Problem-4"
```

```
:at (0 0 7 4 1 1993) (trigger nil nil)        ;;Post an event on the list of things to be done which
                                              ;;will allow report generator to perform a final
                                              ;;audit and generate reports

;;DEFINE THE PARTS:

:CLASS     my-parts
:PARENTS   (parts)                            ;;Define a type of parts and define the parents

  :SUBCLASS    raw-parts                      ;;Define a type of my-parts

      PART-MATERIALS   ((steel 100))          ;;Raw-parts each contain 100 units of steel

       WORKORDER   lathe-raw-parts-wo         ;;Raw-parts each move and are processed as
                                              ;;described by the lathe-raw-parts-wo

  :END-SUBCLASS    raw-parts                  ;;End the definition of raw parts

  :SUBCLASS    lathed-parts                   ;;Define another type of my-parts

  :END-SUBCLASS    lathed-parts               ;;End the definition of lathed parts

  :SUBCLASS    bag-of-waste                   ;;Define another type of my-parts

  :END-SUBCLASS    bag-of-waste               ;;End the definition of bag-of-waste

  :SUBCLASS    compacted-steel-parts          ;;Define still another type of my-parts

  :END-SUBCLASS    compacted-steel-parts      ;;End the definition of compacted-steel-parts
```

EXPLANATION OF SYNTAX

;;DEFINE AVAILABLE RESOURCES WITH UNLIMITED AMOUNTS WHICH CAN BE ADDED ;;TO PROCESSES:

```
:CLASS   my-factory-resources              ;;Define a type of audited, unconstrained
                                           ;;resources
:PARENTS  (resources audited-resources
         PWA-materials)

         audit-period :daily               ;;Instances of my-factory-resources are audited
                                           ;;daily

         instance machine-oil              ;;Create a type of audited unconstrained resources
            epa-hw (F003 F004)             ;;Assign the applicable EPA numbers
            material-class toxic           ;;The material class is of the type toxic
            rcra-listed-material T         ;;The attribute rcra-listed-material is true
            units pints                    ;;The oil comes in pints
```

;;DEFINE AVAILABLE RESOURCES WITH LIMITED AMOUNTS WHICH CAN BE ADDED TO ;;PROCESSES:

```
:CLASS   my-factory-constrained-resources;;Define a type of audited resources whose available
:PARENTS  (resources-with-amounts
            audited-resources)             ;;amounts are limited

         audit-period :daily               ;;Instances of my-factory-constrained-resources are
                                           ;;audited daily
```

EXPLANATION OF SYNTAX

```
:instance kim-wipes                    ;;Create a type of constrained resources called
                                       ;;kim-wipes
  available-resource 1000              ;;There are initially 1000 kim-wipes available

:instance solvent                      ;;Create a type of constrained resources called
                                       ;;solvent
      available-resource 250           ;;There are initially 250 units of solvent available
```

;;DEFINE THE LOCATIONS BETWEEN WHICH PARTS ARE MOVED:

```
:CLASS  my-locations                   ;;Define a type of audited locations
:PARENTS  (locations audited-locations)

   audit-period :daily                 ;;Instances of my-locations are audited daily

  :instance  receiving-location        ;;Create a type of my-locations called receiving-
                                       ;;location

   stored-parts ((raw-parts 3 done))   ;;Receiving  will initially contain three raw parts.
                                       ;;The status "done" indicates that the raw parts can
                                       ;;be moved to another location

:at current-simulation-clock           ;;Post an event on the list of things to be
 decide-operation                      ;;done which will allow the receiving-location to
                                       ;;determine if it should move or process some
                                       ;;material.

 :instance   lathe-location            ;;Create another type of locations called the lathe-
                                       ;;location
```

EXPLANATION OF SYNTAX

```
    attached-workcenters ((lathe-wc))      ;;The lathe-location will contain one piece of
                                           ;;equipment of the type lathe-wc

 :instance   shipping-location             ;;Create another type of locations called the
                                           ;;shipping-location

 :instance   waste-treatment-location      ;;Create another type of locations called the waste-
                                           ;;treatment-location

    stored-bulk ((waste-treatment-vat)     ;;The waste-treatment-location contains two
                                           ;;bulk containers, the waste-treatment-vat and
                 (cleaned-steel-vat))      ;;the cleaned-steel-vat

    attached-workcenters
(waste-treatment-wc waste-treatment-wc
compacting-wc)                             ;;The waste-treatment location contains two
                                           ;;identical pieces of equipment of the type waste-
                                           ;;treatment-wc and one different piece of
                                           ;;equipment of the type compacting-wc

 :instance   waste-storage-location        ;;Create another type of locations called the waste-
                                           ;;storage-location

;;DEFINE THE WORKORDERS WHICH TELL THE LOCATIONS WHERE TO MOVE AND PROCESS
;;THE PARTS

:CLASS    my-workorders                    ;;Define a type of workorders
:PARENTS  (workorders)
  :SUBCLASS   lathe-raw-parts-wo           ;;Define a type of my-workorders
```

EXPLANATION OF SYNTAX

```
workorder-steps                                  ;;This workorder describes two steps

    ((transfer  lathe-location (0 15))           ;;The first step will move the part to
                                                 ;;the lathe location in 15 minutes
(lathe-a-part  lathe-location
          (0  0 8)))                             ;;The second step allows the part to which the
                                                 ;;workorder is attached to be processed by any
                                                 ;;piece of equipment at the lathe location that
                                                 ;;knows how to lathe a part.  Takes 8 hours

   :END-SUBCLASS   lathe-raw-parts-wo            ;;End the subclass

   :SUBCLASS    move-to-shipping-wo              ;;Define another type of my-workorders

   workorder-steps
     ((transfer  shipping-location               ;;This step moves the lathed parts to which
         (0 15 0)))                              ;;the workorder is attached to the shipping
                                                 ;;location in 15 minutes

   :END-SUBCLASS  move-to-shipping-wo            ;;End the definition of move-to-shipping-wo

   :SUBCLASS   move-to-waste-storage-wo          ;;Define another type of my-workorders

   workorder-steps
     ((transfer waste-storage-location           ;;This step moves the part (waste) to which
         (0 0 2)))                               ;;the workorder is attached to the waste-storage-
                                                 ;;location in 2 hours

   :END-SUBCLASS move-to-waste-storage-wo
```

EXPLANATION OF SYNTAX

```
:SUBCLASS   move-to-lathe-location-wo      ;;Define another type of my-workorders

  workorder-steps
    ((transfer lathe-location              ;;This step moves the parts to which
       (0 0 2)))                           ;;the workorder is attached to the lathe-location
                                           ;;in 2 hours

  :END-SUBCLASS move-to-lathe-location-wo
                                           ;;End the definition of move-to-lathe-location-wo

;;DEFINE THE PIECES OF EQUIPMENT THAT WILL PROCESS THE MATERIAL.
;;ALL WORKCENTERS ARE MASS BALANCED - - AMOUNT OF MATERIAL IN THE INPUTS =
;;AMOUNT OF MATERIAL IN THE OUTPUTS:

:CLASS    my-mass-balance-workcenters      ;;Define a type of mass-balance-workcenters
:PARENTS   (mass-balance-workcenters
           audited-workcenters)
   audit-period  :daily                    ;;Instances of my-mass-balance-workcenters
                                           ;;are audited daily

   :SUBCLASS      lathe-wc                 ;;Define the workcenter that will lathe the parts

process-step-descriptions                  ;;Begin defining the steps for processing

;;DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

((FREE                                     ;;This workcenter must be in a free state
                                           ;;to begin processing
```

```
;;DESCRIBE THE INPUTS:

((temp-part   lathe-a-raw-part))      ;;This workcenter can work on a stored-part that
                                      ;;has temp-part as its current workorder entry.
                                      ;;Temp-part is a local name assigned by the user
                                      ;;used only within this process step, allowing
                                      ;;reference to this input by later sections.

()                                    ;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

((machine-oil 5))                     ;;This workcenter requires 5 units of machine oil
                                      ;;from the stockroom. The machine oil will be
                                      ;;incorporated into the process output

;;REUSABLE RESOURCES:

()                                    ;;This workcenter requires no reusable resources

;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:

(FREE                                 ;;This workcenter will be left in a free state at
                                      ;;the end of this processing step

steel                                 ;;The workcenter will distribute any materials,
                                      ;;not specifically described in its output mass
                                      ;;balance, relative to the amount of steel in each
                                      ;;output

(:new lathed-parts                    ;;Create a new instance of lathed-parts
```

EXPLANATION OF SYNTAX

```
   move-to-shipping-wo                  ;;The name of the workorder to be attached to
                                        ;;the lathed-parts

   ()                                   ;;The part attributes of lathed-parts

(steel  .99))                           ;;Place 99% of the steel from the input into a
                                        ;;new instance of lathed parts

(:feed    waste-treatment-location      ;;Send to a bulk container in the waste-treatment-
                                        ;;location

waste-treatment-vat                     ;;The name of the bulk container

(steel .01)                             ;;Put 1% of the steel into the bulk container

(machine-oil 1.0)))))                   ;;Put 100% of the machine oil into the bulk
                                        ;;container

 :END-SUBCLASS   lathe-wc               ;;End the definition of lathe-wc

 :SUBCLASS  waste-treatment-wc          ;;Define the workcenter that will
                                        ;;process material from the waste-treatment-vat

process-step-descriptions               ;;Begin defining the steps for processing
                                        ;;the material (waste)
```

;;DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

```
((FREE                                  ;;This workcenter must be in a free state
                                        ;;to begin processing
```

EXPLANATION OF SYNTAX

DESCRIBE THE INPUTS. THIS TIME, INPUT FROM A BULK CONTAINER RATHER THAN A PART:

((**:bulk** waste-treatment-vat	;;The name of the bulk container from which to ;;draw material
`steel`	;;The workcenter will draw material relative to ;;the amount of steel in the vat
`2`	;;The number of units of the basis material that ;;goes into each batch
`nil`	;;The amount of material that can be processed in ;;the time specified below -- nil or a number
`(0 0 2)`	;;It will take two hours to process the amount
`2))`	;;The minimum batch size
()	;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

`((kim-wipes 5) (solvent 24))`	;;This workcenter requires 5 units of ;;kim-wipes and 24 units of solvent from the ;;stockroom (resources). Both the kim-wipes ;;and the solvent will be incorporated into the ;;process output

EXPLANATION OF SYNTAX

;;REUSABLE RESOURCES:

```
()                                  ;;This workcenter requires no reusable resources
```

;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:

```
(FREE                               ;;This workcenter will be left in a free state at
                                    ;;the end of this processing step

steel                               ;;The workcenter will distribute any materials,
                                    ;;not specifically described in its output mass
                                    ;;balance, relative to the amount of steel in each
                                    ;;output

(:new bag-of-waste                  ;;Create a new instance of bag-of-waste parts

  move-to-waste-storage-wo          ;;The name of the workorder to be attached to
                                    ;;the bag-of-waste

()                                  ;;The part attributes of the bag-of-waste

(kim-wipes 1.0)                     ;;Send all the kim-wipes,

(solvent  1.0)                      ;;all the solvent,

(machine-oil 1.0))                  ;;all the machine oil into a part (bag-of-waste)

(:bulk  cleaned-steel-vat           ;;Send to a bulk container named cleaned-steel-vat

(steel 1.0)))))                     ;;all of the steel
```

EXPLANATION OF SYNTAX

:END-SUBCLASS waste-treatment-wc	;;End the definition of waste-treatment-wc
:SUBCLASS compacting-wc	;;Define the workcenter that will process material ;;from the cleaned-steel-vat
process-step-descriptions	;;Begin defining the steps for the compacting ;;process of the material

;;DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

((FREE	;;This workcenter must be in a free state ;;to begin processing

;;DESCRIBE THE INPUTS. THIS TIME, INPUT FROM A BULK CONTAINER RATHER
;;THAN A PART:

((**:bulk** cleaned-steel-vat	;;The name of the bulk container from which to ;;draw the material
total mass	;;The workcenter will draw an absolute amount of ;;material
1.0	;;The number of units of material to draw
nil	;;The amount of material that can be processed in ;;time defined below -- specify nil or a number ;;from the cleaned-steel-vat. The batch size ;;is one unit of the total mass of all materials

EXPLANATION OF SYNTAX

(0 0 2)	;;The amount of time to process the amount ;;is 2 hours
1.0))	;;The minimum batch size
()	;;Place holder for future expansion of the model
;;MATERIAL RESOURCES:	
()	;;This workcenter requires no material resources.
;;REUSABLE RESOURCES:	
()	;;This workcenter requires no reusable resources
;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:	
(FREE	;;This workcenter will be left in a free state at ;;the end of this processing step
()	;;There is no primary material with which to ;;distribute the outputs for this processing step
(:**new** compacted-steel-parts	;;Create a new instance of compacted-steel-parts
move-to-lathe-location-wo	;;The name of the workorder to be attached to ;;the new part
()	;;The part attributes of the new part

```
(steel 1.0)))))                    ;;Put all of the steel into the new part

:END-SUBCLASS  compacting-wc       ;;End the definition of the compacting-wc
```

EM_PROBLEM 5

EDITING PROBLEM 4, USE PWA TO MODEL THE FOLLOWING:

1) CHANGE THE LATHE-WORKCENTER SO THAT IT CAN WORK EITHER ON RAW PARTS OR COMPACTED-STEEL-PARTS.

2) THE RESOURCE NEEDED FOR LATHING COMPACTED-STEEL-PARTS IS 10 UNITS OF MACHINE OIL.RATHER THAN 5 UNITS FOR RAW-PARTS.

3) THE NEW LATHED-COMPACTED-STEEL-PARTS CONTAIN 99% OF THE STEEL AND GET TRANSFERRED TO SHIPPING.

4) THE REMAINING 1% OF THE STEEL AND 100% OF MACHINE-OIL WILL BE SENT TO THE WASTE-TREATMENT-VAT IN THE WASTE-TREATMENT-LOCATION

Learning Objectives:
Learn how to model the processing so that a workcenter can perform two different tasks.

EXPLANATION OF SYNTAX

;;SET THE CLOCK - HOW LONG DO I WANT TO RUN THIS SIMULATION?

```
:CLOCK   (0.0 0 8 1 1 1993)      ;;Start the simulation on January 1, 1993 at 8:00 am
:LIMIT   (0.0 0 8 4 1 1993)      ;;Run the simulation for 3 days
```

EXPLANATION OF SYNTAX

```
:CLASS    my-foremen                          ;;Define a type of foremen that will both audit
:PARENTS (reporter-foremen auditing-foremen)  and generate reports when triggered

  :instance  report-generator                 ;;report-generator will generate reports
                                              ;;according to the following list of report
                                              ;;types and report file names:

   processing-cumulative   "CUMULATIVE-Problem-5"
   processing-incremental  "INCREMENTAL-Problem-5"
    feed                   "FEED-REPORT-Problem-5"
   materials              "MATERIALS-Problem-5"
   material-handling       "MATERIAL-HAND-Problem-5"
   part-handling           "PART-HANDLING-Problem-5"

:at (0 0 7 4 1 1993) (trigger nil nil)        ;;post an event on the list of things to be done which
                                              ;;will allow report-generator to perform its final
                                              ;;audit and generate its reports
```

EXPLANATION OF SYNTAX

;;TYPES OF PMS REPORTS

processing-incremental	;;Generates a table of the incremental ;;processing data
feed	;;Generates a table showing the sources of ;;all material into each location
materials	;;Generates a report showing the materials ;;holding for each location
material handling	;;Generates a table of the material inputs, ;;additions, holdings and outputs for each ;;location
part-handling	;;Generates a table of parts handled by each ;;location
ascii	;;Generates a file to be used as input to a ;;graphics package or spreadsheet

EXPLANATION OF SYNTAX

;;DEFINE THE PARTS TO BE PROCESSED:

```
:CLASS     my-parts                          ;;Define a type of part and define the parents
:PARENTS (parts)

  :SUBCLASS   raw-parts                      ;;Define a type of my-parts

      PART-MATERIALS   ((steel 100))         ;;Raw-parts each contain 100 units of steel

      WORKORDER   lathe-raw-parts-wo         ;;Raw-parts each move and are processed as
                                             ;;described by the lathe-raw-parts-wo

  :END-SUBCLASS   raw-parts                  ;;End the definition of raw parts

  :SUBCLASS   lathed-parts                   ;;Define the subclass of parts that get lathed

  :END-SUBCLASS   lathed-parts               ;;End the definition of lathed parts

  :SUBCLASS   lathed-compacted-parts         ;;Define the subclass of lathed compacted parts

  :END-SUBCLASS   lathed-compacted-parts     ;;End the definition of lathed compacted parts

  :SUBCLASS   bag-of-waste                   ;;Define the subclass of the bag of waste

  :END-SUBCLASS   bag-of-waste               ;;End the definition of bag of waste
```

EXPLANATION OF SYNTAX

```
:SUBCLASS    compacted-steel-parts        ;;Define the subclass of compacted steel parts

:END-SUBCLASS  compacted-steel-parts      ;;End the definition of compacted steel parts
```

;;DEFINE THE RESOURCES TO BE ADDED TO A PROCESS BY A WORKCENTER:

```
:CLASS  my-factory-resources              ;;Define a type of resource that also audits
:PARENTS  (resources                      ;;resource usage
  audited-resources PWA-materials)

        audit-period :daily               ;;Instances of my-factory-resources.
                                          ;;Each conduct their audit on a daily basis

     :instance machine-oil                ;;Create a type of unconstrained resources
          epa-hw (F003 F004)              ;;Assign the applicable EPA numbers
          material-class toxic            ;;The material class is of the type toxic
          rcra-listed-material T          ;;The attribute rcra-listed-material is true
          units pints                     ;;The oil comes in pints

:CLASS  my-factory-constrained-resources;;Define a type of resource whose available
:PARENTS  (resources-with-amounts         ;;amounts are constrained
               audited-resources)

        audit-period :daily               ;;Instances of my-factory-constrained-resources.
                                          ;;Each conduct their audit on a daily basis

     :instance kim-wipes                  ;;Create a resource whose available amounts
                                          ;;are constrained resource called kim-wipes
```

EXPLANATION OF SYNTAX

```
  available-resource 1000                ;;Kim-wipes will initially have 1000 units
                                         ;;available

          instance solvent                ;;Create a resource whose available amounts
                                         ;;are constrained resource called solvent

         available-resource 250          ;;Solvent will initially have 250 units
                                         ;;available
```

;;DEFINE THE LOCATIONS BETWEEN WHICH PARTS ARE MOVED:

```
:CLASS  my-locations                     ;;Define a type of location
:PARENTS  (locations audited-locations)

      audit-period :daily                ;;Instances of my-locations.
                                         ;;Each conduct their audit on a daily basis

     :instance  receiving-location       ;;Create a location called receiving

     stored-parts ((raw-parts 3 done))    ;;Receiving  will initially contain three raw parts.
                                         ;;The status "done" indicates that they can
                                         ;;be moved to another location

:at current-simulation-clock             ;;Post an event on the list of things to be
 decide-operation                        ;;done which will allow receiving to
                                         ;;determine if it should move or process
                                         ;;some material.
```

EXPLANATION OF SYNTAX

```
:instance    lathe-location                  ;;Create another location called the lathe location

    attached-workcenters ((lathe-wc))        ;;The lathe location will contain one piece of
                                             ;;equipment of the type lathe-wc

:instance    shipping-location               ;;Create another location called the shipping location

:instance    waste-treatment-location        ;;Create another location called the waste-
                                             ;;treatment-location

   stored-bulk ((waste-treatment-vat)        ;;The waste-treatment-location will contain one
      (cleaned-steel-vat))                   ;;bulk container named waste-treatment-vat

       attached-workcenters
 (waste-treatment-wc waste-treatment-wc
        compacting-wc)                       ;;The waste-treatment location will contain two
                                             ;;pieces of equipment of the type waste-
                                             ;;treatment-wc and compacting-wc

:instance    waste-storage-location          ;;Create another location called the waste-
                                             ;;storage-location

;;DEFINE THE WORKORDERS WHICH TELL THE LOCATIONS WHERE TO MOVE THE
;;PARTS AND HOW TO PROCESS THE PARTS

:CLASS    my-workorders                      ;;Define a type of workorder
:PARENTS  (workorders)
```

	EXPLANATION OF SYNTAX
:SUBCLASS lathe-raw-parts-wo	;;Define a type of my-workorders
workorder-steps	;;This workorder describes two steps.
((**transfer** lathe-location (0 15))	;;The first step will transport the part to ;;which the workorder is attached to the lathe ;;location in 15 minutes
(*lathe-a-part* lathe-location (0 0 8)))	;;The second step allows the part to which the ;;workorder is attached to be processed by any ;;piece of equipment at the lathe location that ;;knows how to lathe a part
:END-SUBCLASS lathe-raw-parts-wo	;;End the definition of lathe-raw-parts-wo
:SUBCLASS move-to-shipping-wo	;;Define another type of my-workorders
workorder-steps	
((**transfer** shipping-location (0 15 0)))	;;This step transfers the lathed parts to which ;;the workorder is attached to the shipping ;;location in 15 minutes
:END-SUBCLASS move-to-shipping-wo	;;End the definition of move-to-shipping-wo
:SUBCLASS move-to-waste-storage-wo	;;Define another type of my-workorders
workorder-steps	
((**transfer** waste-storage-location (0 0 2)))	;;This step transfers the waste to which ;;the workorder is attached to the waste-storage- ;;location in 2 days

EXPLANATION OF SYNTAX

```
:END-SUBCLASS move-to-waste-storage-wo
                                        ;;End the definition of move-to-shipping-wo

:SUBCLASS   move-to-lathe-location-wo ;;Define another type of my-workorders

workorder-steps
  ((transfer lathe-location             ;;This step transfers the parts to which
      (0 0 2))                          ;;the workorder is attached to the lathe
                                        ;;location in 2 days
  (lathe-a-compacted-part
    lathe-location   (0 30)))           ;;This step lathes the compacted
                                        ;;part in the lathe location in 30 minutes

:END-SUBCLASS move-to-lathe-location-wo
                                        ;;End the definition of move-to-lathe-location-wo

;;DEFINE THE PIECES OF EQUIPMENT THAT WILL PROCESS THE MATERIAL
;;ALL WORKCENTERS ARE MASS BALANCED (AMOUNT OF MATERIAL IN THE INPUTS =
;;AMOUNT OF MATERIAL IN THE OUTPUTS)

:CLASS    my-mass-balance-workcenters   ;;Define a type of mass balance workcenters
:PARENTS   (mass-balance-workcenters
         audited-workcenters)

   audit-period  :daily                 ;;Instances of my-mass-balance-workcenters
                                        ;;each conduct their audit on a daily basis

   :SUBCLASS       lathe-wc             ;;Name the workcenter that will perform the
                                        ;;process of lathing the parts
```

	EXPLANATION OF SYNTAX
process-step-descriptions	;;Begin defining the steps for processing ;;the part

;;MY-MASS-BALANCE-WC CAN DO TWO DIFFERENT PROCESSING STEPS, REPRESENTED
;;BY TWO DIFFERENT PROCESS STEP DESCRIPTIONS:

;;DESCRIBE THE INITIAL STATE OF THE EQUIPMENT:

((FREE	;;This workcenter must be in a free state ;;ready to process

;;DESCRIBE THE INPUTS:

((*temp-part* lathe-a-raw-part))	;;This workcenter can work on a stored part ;;that has lathe-a-raw-part as its current ;;workorder entry. *Temp-part* is a local name ;;assigned by the user used only within this process ;;step, allowing reference to this input by later ;;sections.
()	;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

((machine-oil 5))	;;This process step requires 5 units of ;;machine oil from the stockroom ;;(resources). This machine oil will be ;;incorporated into the process output

EXPLANATION OF SYNTAX

;;REUSABLE RESOURCES:

```
()                                      ;;This workcenter requires no reusable resources
```

;;BEGIN DESCRIBING THE OUTPUTS OF THE WORKCENTER:

```
(FREE                                   ;;This workcenter will be left in a free state at
                                        ;;the end of this processing step

steel                                   ;;The workcenter will distribute any materials,
                                        ;;not specifically described in its output mass
                                        ;;balance, relative to the amount of steel in each
                                        ;;output

(:new lathed-parts                      ;;the new output
move-to-shipping-wo                     ;;the work-order attached to the new output
()                                      ;;no attributes described here

                                        ;;Place 99% of the steel from the input into a
(steel .99))                            ;;new instance of lathed compacted parts

(:feed  waste-treatment-location
   waste-treatment-vat (steel.01)
      (machine-oil 1.0))))              ;;Send 1% of the steel and 100% of the
                                        ;;machine oil to the waste-treatment-vat

(FREE                                   ;;The workcenter must begin in a free state

((temp-part  lathe-a-compacted-part))   ;;This workcenter can work on a stored part
                                        ;;that has lathe-a-compacted-part as its current
                                        ;;workorder entry. Temp-part is a local name
```

```
                                        ;;assigned by the user used only within this process
                                        ;;step, allowing reference to this input by later
                                        ;;sections.

()                                      ;;Place holder for future expansion of the model

((machine-oil 10))                      ;;This step requires 10 units of the
                                        ;;unconstrained resource machine-oil

()                                      ;;No reusable resources

(free                                   ;;The workcenter must end in a free state

steel                                   ;;primary material placeholder

(:new lathed-compacted-parts  move-to-
  shipping-wo  ()  (steel  .99))        ;;The new lathed compacted parts comprised
                                        ;;of 99% of the steel will be
                                        ;;moved to shipping in 15 minutes

(:feed waste-treatment-location
  waste-treatment-vat  (steel .01)
  (machine-oil 1.0)))))                 ;;The remaining 1% of steel and 100% of
                                        ;;the resource machine-oil will be sent
                                        ;;to the waste treatment vat in the
                                        ;;waste treatment location

:END-SUBCLASS   lathe-wc                ;;End the definition of lathe-wc
```

EXPLANATION OF SYNTAX

```
:SUBCLASS   waste-treatment-wc          ;;define a type of my-mass-balance-workcenters
process-step-descriptions

((FREE                                  ;;The workcenter must be in a free state

   ;; container-name      batch-basis   batch-size   amt   (per)   time  min

((:bulk waste-treatment-vat     steel        2          nil   (0 0 2)     2))

                                        ;;This workcenter can work on a batch of material
                                        ;;from the waste-treatment-vat.  The batch size
                                        ;;is two units of steel. The batch time is 2 hours.

()                                      ;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

((kim-wipes 5) (solvent 24))            ;;This workcenter requires 5 units of
                                        ;;kim-wipes and 24 units of solvent from the
                                        ;;stockroom (resources).  Both the kim-wipes
                                        ;;and the solvent will be incorporated into the
                                        ;;process output

;;REUSABLE RESOURCES:

()                                      ;;This workcenter requires no reusable resources
```

EXPLANATION OF SYNTAX

;;DESCRIBE THE OUTPUTS OF THE WORKCENTER:

(FREE ;;This workcenter will be left in a free state at
;;the end of this processing step

steel ;;The workcenter will distribute any materials,
;;not specifically described in its output mass
;;balance, relative to the amount of steel in each
;;output

new class of parts	new workorder	part-attributes	mass-balance
(:**new** bag-of-waste	move-to-waste-storage-wo	()	(kim-wipes 1.0) (solvent 1.0) (machine-oil 1.0))

;;Place 100% of the kim-wipes, solvent and machine
;;oil into a new place called bag-of-waste and apply
;;the current workorder to move it to the waste
;;storage location.

(:**bulk** cleaned-steel-vat (steel 1.0)))))

;;Place 100% of the steel into a bulk container called
;;cleaned-steel-vat

:END-SUBCLASS waste-treatment-wc ;;End the definition of waste-treatment-wc

:SUBCLASS compacting-wc ;;Name the workcenter that will perform the
;;process of compacting the recycled steel

process-step-descriptions Begin defining the steps for the compacting process.

BEGIN BY DESCRIBING THE INITIAL STATE OF THE EQUIPMENT:

((FREE ;;This workcenter must be in a free state
;;-- ready to process

;;DESCRIBE THE INPUTS. THIS TIME, INPUT FROM A BULK CONTAINER RATHER
;;THAN A PART:

container-name	batch-basis	batch-size	amt	(per)	time min
((**:bulk** cleaned-steel-vat	steel	1.0	nil	(0 0 2)	1.0))

;;This workcenter can work on a batch of material
;;from the cleaned-steel-vat. The batch size
;;is one unit of the steel in the tank. The batch time
;;2 hours.

() ;;Place holder for future expansion of the model

;;MATERIAL RESOURCES:

() ;;This workcenter requires no resources.

;;REUSABLE RESOURCES:

```
()                                          ;;This workcenter requires no reusable resources
```

;;BEGIN DESCRIBING THE OUTPUTS OF THE WORKCENTER:

```
(FREE                                       ;;This workcenter will be left in a free state at
                                            ;;the end of this processing step

()                                          ;;There is no primary material for this processing
                                            ;;step.

;;new class of parts      new workorder      part-attributes      total-mass

(:new compacted-steel-parts move-to-lathe-
                            location-wo            ()              (steel 1.0)))))

                                            ;;Place 100% of the steel into a new place called
                                            ;;compacted steel parts and apply the current
                                            ;;workorder to move it to the lathe location

:END-SUBCLASS  compacting-wc                ;;End the definition of compacting-wc
```

```
Initialize from base scenario file Macintosh HD:Desktop Folder:Disk [redacted]
SP
OCK "Jan  1, 1993 08:00"
MIT "Jan  8, 1993 08:00"
LASS OUR-PARTS
NTS (PARTS)
ASS RAW-PARTS
MATERIALS ((STEEL 100))
RKORDER LATHE-RAW-PARTS-WO
ND-SUBCLASS RAW-PARTS
UBCLASS LATHED-PARTS
ND-SUBCLASS LATHED-PARTS
LASS MY-FACTORY-RESOURCES
ARENTS (RESOURCES)
NSTANCE MACHINE-OIL
LASS MY-LOCATIONS
ARENTS (LOCATIONS)
NSTANCE RECEIVING
ORED-PARTS ((RAW-PARTS 3 DONE))
T CURRENT-SIMULATION-CLOCK DECIDE-OPERATION
NSTANCE LATHE-LOCATION
TACHED-WORKCENTERS ((LATHE-WC))
NSTANCE SHIPPING-LOCATION
NSTANCE WASTE-TREATMENT-LOCATION
ORED-BULK ((WASTE-TREATMENT-VAT))
LASS MY-WORKORDERS
ARENTS (WORKORDERS)
UBCLASS LATHE-RAW-PARTS-WO
RKORDER-STEPS ((TRANSFER LATHE-LOCATION (0 15)) (LATHE-A-PART LATHE-LOCATION (0 0 8)))
ND-SUBCLASS LATHE-RAW-PARTS-WO
UBCLASS MOVE-TO-SHIPPING-WO
RKORDER-STEPS ((TRANSFER SHIPPING-LOCATION (0 15 0)))
ND-SUBCLASS MOVE-TO-SHIPPING-WO
S MY-MASS-BALANCE-WORKCENTERS
NTS (MASS-BALANCE-WORKCENTERS)
UBCLASS LATHE-WC
ROCESS-STEP-DESCRIPTIONS ((FREE ((TEMP-PART LATHE-A-PART)) NIL ((MACHINE-OIL 5)) NIL (FR
E NIL (:NEW LATHED-PARTS MOVE-TO-SHIPPING-WO NIL (STEEL 0.99)) (:FEED WASTE-TREATMENT-LOC
TION WASTE-TREATMENT-VAT (STEEL 0.01) (MACHINE-OIL 1.0)))))
END-SUBCLASS LATHE-WC
EXIT
reate classes (VISIBLE-SIMULATION-OBJECTS)
reate instances (VISIBLE-SIMULATION-OBJECTS)
nitialize instances (VISIBLE-SIMULATION-OBJECTS)
```

```
Post initial events (VISIBLE-SIMULATION-OBJECTS)
Done

Jan  1, 1993 08:00 RECEIVING     -> DECIDE-OPERATION
Jan  1, 1993 08:00 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 08:00 RECEIVING     -> BEGIN-TRANSFER #<RAW-PARTS-6>
Jan  1, 1993 08:00 RECEIVING     -> BEGIN-TRANSFER #<RAW-PARTS-4>
Jan  1, 1993 08:00 RECEIVING     -> DECIDE-OPERATION
Jan  1, 1993 08:00 RECEIVING     -> BEGIN-TRANSFER #<RAW-PARTS-5>
Jan  1, 1993 08:00 RECEIVING     -> DECIDE-OPERATION
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-4>
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-5>
Jan  1, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 08:15 LATHE-LOCATION -> END-TRANSFER #<RAW-PARTS-6>
Jan  1, 1993 08:15 LATHE-WC-4    -> BEGIN-PROCESSING
Jan  1, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 LATHE-WC-4    -> END-PROCESSING
Jan  1, 1993 16:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-4> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  1, 1993 16:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 LATHE-WC-4    -> BEGIN-PROCESSING
Jan  1, 1993 16:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-6>
Jan  1, 1993 16:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  1, 1993 16:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-6>
Jan  1, 1993 16:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-WC-4    -> END-PROCESSING
Jan  2, 1993 00:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-4> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  2, 1993 00:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-WC-4    -> BEGIN-PROCESSING
Jan  2, 1993 00:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-7>
Jan  2, 1993 00:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 00:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-7>
Jan  2, 1993 00:30 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 LATHE-WC-4    -> END-PROCESSING
Jan  2, 1993 08:15 WASTE-TREATMENT-LOCATION -> RECEIVE-FEED #<LATHE-WC-4> WASTE-TREATMENT
-VAT ((MACHINE-OIL 5.0) (STEEL 1.0))
Jan  2, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 WASTE-TREATMENT-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 SHIPPING-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:15 LATHE-LOCATION -> BEGIN-TRANSFER #<LATHED-PARTS-8>
Jan  2, 1993 08:15 LATHE-LOCATION -> DECIDE-OPERATION
Jan  2, 1993 08:30 SHIPPING-LOCATION -> END-TRANSFER #<LATHED-PARTS-8>
Jan  2, 1993 08:30 SHIPPING-LOCATION -> DECIDE-OPERATION
?
```

APPENDIX C

FRED Commands

COMMON LISP TEXT EDITOR FOR THE MACINTOSH

FRED

Macintosh Common Lisp includes a powerful text editor called FRED. You will use FRED to write your scenarios. Fred is a very close cousin of EMACS, which is the common lisp text editor used on the IBM and SUN computer platforms.

Aside from being versatile, FRED is easy-to-use and includes the standard Macintosh editor features, such as multiple windows, mouse-based cut and paste operations, and so on. If you are an experienced Macintosh user, you can get started with FRED simply by using the editing commands you already know.

The FRED commands designed specifically for Lisp programming are summarized below into the following functions:

Commands for manipulating lisp

Commands for maximizing use of the Macintosh windows environment

Commands for inspecting functions and getting help

Commands for specialized editing.

Fred commands use two modifier keys, called *Control* and *Meta*

To type Control modifiers use the Control key. If you keyboard does not have the control key, use Command or Command-Shift, depending on the value of the variable *emacs-mode*

To type Meta modifiers on the Macintosh keyboard, use the Option key.

FRED

SPECIALIZED LISP MANIPULATION COMMANDS:

ENTER

Press on the numeric keyboard to evaluate Lisp expressions from Fred windows.

PARENTHESIS

To automatically balance parenthesis in Lisp expressions, place the insertion point after a close parenthesis, or before an open parenthesis. This will cause the matching parenthesis to blink and will assist you to easily detect errors (missing parenthesis or adding too many parenthesis). Double-clicking with the mouse after a close parenthesis, or before an open parenthesis highlights forward or backward to the corresponding matched parenthesis.

TAB

Pressing tab after a return indents the new line appropriately

CONTOL META Q

Reindents the current Lisp expression

META-close parenthesis

Moves the cursor into position for typing the next expression

FRED COMMANDS

Select this from the tools menu for a window of all FRED commands. You can also print this window for easy reference.

BASIC COMMANDS FOR WRITING YOUR SCENARIO FILES IN FRED

When you select the PWA-1.0 icon from your Macintosh hard-disk, you will automatically enter the exciting and interactive environment of PWA. Written in Macintosh Common Lisp, PWA on the Maintosh consists of the following main menu items:

FILE EDIT EVAL TOOLS WINDOWS SIMULATION

THE FILE SUB-MENU

This menu contains all of the standard Macintosh file commands such as File OPen, File New, File Save etc. Also use this menu to print your files and/or windows.

THE EDIT SUB-MENU

This menu contains all of the standard Macintosh editing features which you can use to edit your scenario files in PWA-1.0. Cut, Copy, Paste can be used to duplicate repetitive sections of long scenario files and can save on coding time!

THE EVAL SUB-MENU

This menu is a special Lisp menu that works in concert with both FRED and the Listener windows that you use to evaluate a file or buffer. Also use this menu to load and compile a file.

THE TOOLS SUB-MENU

This menu contains a number of Lisp aide programs which you can use to facilitate your interactive programming sessions in PWA. The most commonly used ones you will employ are the Fred and Listener commands you can generate, reference and even print out for facile reference.

THE WINDOWS SUB-MENU

This menu conveniently tells you what the active window is with a checkmark.

THE SIMULATION SUB-MENU

This is the menu you select to load and run your simulations.

COMMONLY USED COMMANDS

REMEMBER, THE COMMAND KEY IS THE LOWER LEFT HAND KEY WITH THE PICTURE OF THE APPLE ON IT!

Command Keys

Command (.)	Cancel errors or abort from evaluation menu
Command p	Print a file
Command S	Save a buffer to a file
Command O	Open up a file
Command E	Automatically evaluate a line of code. (You can also click after the end of the expression)

Control Keys

Control K	Delete a line of code and send it to the Lisp kill buffer
Control Y	Yank out a deleted expression from the kill buffer
Control S	Search for a designated string of text in the file

File commands

The following commands control files:

New (⌘-N) Creates an editor window for a new file.

Open... (⌘-O) Allows you to select a text file and creates a new editor window for the file.

Open Selected File (⌘-D)
If there's a selection in the top editor window, Macintosh Common Lisp attempts to parse this selection as a pathname and to create an editor window for the pathname's file.

Close (⌘-W) Closes the current window.

If the window is not a Listener, and if its contents have been edited since the last time it was saved, Macintosh Common Lisp displays a dialog box asking you if you want to save or throw away the changes.

Save (⌘-S) Saves the contents of the active window to the file named in the title bar of the window. If the window isn't associated with a disk file, Save As is invoked.

Save As... Allows you to specify a directory and filename, and saves the contents of the active window to that directory/filename.

Save Copy As... Allows you to specify a directory and filename, and saves a copy of the contents of the active window to that directory/filename. Save Copy As is often used to save the contents of the Listener to a file.

Revert (⌘-R) Reverts to the version of the window contents last saved to disk. Before the reversion occurs, you're asked to verify whether you really want to revert to the last version saved.

Page Setup... Allows you to set printing options for the current printer.

Print... (⌘-P) Prints the contents of the active window on the currently selected printer.

Quit (⌘-Q) Closes all windows and exits the Lisp environment. If any window contains revisions that have not been saved to disk, you're given the option of saving them.

Unbound variable

If you get the error message "`Unbound variable` *symbol-name*", here are some possible explanations and solutions:

- You may have mistyped the symbol.

  Check it and try again.
- You may have defined the symbol locally, within a function, rather than as a global symbol.
- The symbol may be defined in another package.

△ **Important** The package problem occurs frequently in conversions from the older Common Lisp standard. The new draft standard of Common Lisp uses the package name `common-lisp`, whereas older code uses the package name `lisp`. △

No applicable primary method for args

If you get the error message "`No applicable primary method for args to` *function*", here are a possible explanation and solution:

- The function you are using is a generic function, which has different methods for different classes of arguments. It does not have a method for the class of the argument you are using. For example, you may have applied a function to a button dialog item that has methods only for table dialog items.

  Use another function or write a method. See *Common Lisp: The Language* and Appendix C of the *Macintosh Common Lisp Reference* for further information on generic functions and methods.

Appendix B Common Errors and What to Do About Them

Contents

Errors / 88
Nothing happens after you type an expression and press Return / 88
File filename does not exist / 88
Unbound variable / 89
No applicable primary method for args / 89
Undefined function called with arguments / 90
Wrong number of arguments / 90
Value is not of the expected type / 90

Every programmer has a personal set of favorite errors, and the following list is not comprehensive. However, a few errors appear frequently. This appendix lists the ones that appear most often and shows how to recover from them.

If you are having trouble with an error, use the debugging techniques explained in Chapter 4 and look at the syntax of the functions you are using.

If you still can't fix the error, talk to a fellow Macintosh Common Lisp programmer. One way of doing that is to post the error on `info-mcl` (see Chapter 5, "Where to Go From Here"). Please include code samples if possible.

Incremental search

A search is said to be incremental when the searching process begins as soon as you begin to enter the search string. For instance, if you are searching for the word `defclass`, Macintosh Common Lisp finds the first `d` as soon as you type it. When you add `e`, it finds the first occurrence of `de`, and so on. This provides autocompletion for the search process.

Full details on searching can be found in the section "Incremental Searching in Fred" in "Editing in Macintosh Common Lisp," Chapter 2 of the *Macintosh Common Lisp Reference*.

The keystroke sequences in Table A-8 are used during incremental searches.

■ **Table A-8 Search command keystrokes**

Function	Keystroke
Searches forward incrementally	Control-S
Searches backward incrementally	Control-R
Deletes characters from search string	Delete
Terminates incremental search	Escape
Cancels incremental search	Control-G
Inserts quoted character	Control-Q
Copies word following insertion point into search string	Control-W
Copies line following insertion point into search string	Control-Y

Deletion

The sequences shown in Table A-5 are used to delete text and spaces.

■ **Table A-5** Deletion command keystrokes

Function	Keystroke
Deletes character to left of insertion point	Delete
Deletes word to left of insertion point	Meta-Delete
Deletes expression to left of insertion point	Control-Meta-Delete
Deletes character to right of insertion point	Control-D, Forward Delete on Apple Extended Keyboard
Deletes word or part of word to right of insertion point, adds to kill ring	Meta-D
Deletes from insertion point to end of line, adds to kill ring	Control-K
Deletes expression to right of insertion point, adds to kill ring	Control-Meta-K
Deletes current selection, adds to kill ring	Control-W
Copies current selection onto kill ring without deleting	Meta-W
Deletes whitespace (spaces or tabs) from insertion point to next nonwhite character	Control-X Control–Space bar
Deletes whitespace (spaces or tabs) around insertion point, adds one space	Meta–Space bar

Selection

The keystroke sequences shown in Table A-3 are used to select text.

■ **Table A-3** Selection command keystrokes

Function	Keystroke
Selects current expression	Control–Meta–Space bar
Selects current top-level expression (the expression that has an open parenthesis flush with the left margin)	Control-Meta-H
Selects entire buffer	Control-X H

Insertion

The sequences shown in Table A-4 are used to insert text and space.

■ **Table A-4** Insertion command keystrokes

Function	Keystroke
Inserts new line without moving insertion point	Control-O
Reindents current line or selection	Tab
Reindents current expression	Control-Meta-Q
Inserts Return followed by Tab	Control-Return
Yanks (pastes) current kill-ring string	Control-Y
Rotates the string that is pasted from the kill ring	Meta-Y

(continued)

Help commands

The help functions are bound to the keystroke sequences shown in Table A-1.

■ **Table A-1** Help command keystrokes

Function	Keystroke
Displays Fred Help window with list of all keyboard commands	Control-?
Displays definition of current expression	Meta-period
Inspects current expression	Control-X Control-I
Prints argument list of current expression	Control-X Control-A
Prints documentation string of current expression	Control-X Control-D
Prints information about current Fred window	Control-=

5. **Open a Fred file.**

   Choose New from the File menu, or open an existing file and select where you would like the copied information to go.

6. **"Yank" the information from the kill ring by pressing Control-Y, the Fred Paste command keystroke.**

Since Fred produces plain ASCII text, you can also import files from Fred into other environments such as word processors and databases.

You can read more about Fred in Appendix A of this book, "Fred Commands," and in Chapters 2 and 14 of the *Macintosh Common Lisp Reference*.

Getting help on Listener commands

A few commands work only in the Listener. The Listener Commands window lists all the Listener commands that differ from Fred commands, and their command keystrokes.

What you've learned about the Macintosh Common Lisp environment

You've learned that there are two basic windows in the Macintosh Common Lisp environment, the Listener and the editor. First you created a Listener window and evaluated code in it. Then you added a Fred window and edited a buffer (an editable copy of a file). You created a small program in the buffer and saved the buffer to a file. You've used some Fred commands and found out how to get help on Fred commands.

Remember, you can take a break at any time and later pick up where you left off. To get out of Macintosh Common Lisp, simply choose Quit from the File menu.

When you go on, you'll learn how to compile files, load and use the Interface Toolkit, do quick prototypes of your own dialogs and menus, and debug coding errors.

Getting help on Fred commands

The Fred Commands window (Figure 2-8) lists all the commands currently available in Fred and their command keystrokes. It can be invoked by pressing Control-/ or Control-?.

If you know the name of the command you want, you can search for its command keystroke in this window, using the Fred search command, Control-S.

Here are some useful strings to search on when you don't know the name of the comman..

EVAL (evaluate)

SEXP (symbolic expression; Lisp expression)

KILL (add to the kill ring)

CHAR (character)

SCREEN

REGION

WINDOW

BUFFER

WORD

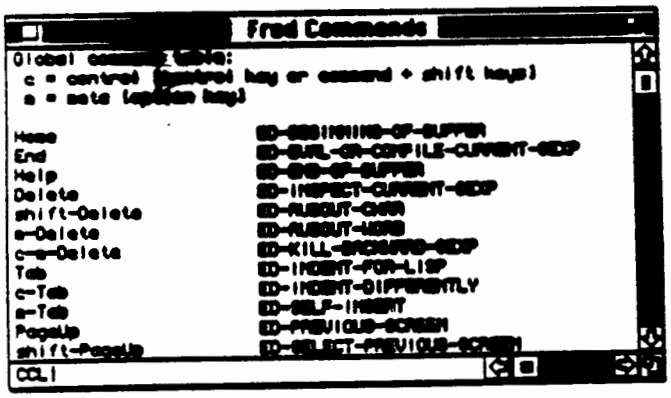
COMPILE

DELETE

INSPECT

MACROEXPAND

■ **Figure 2-8 The Fred Commands window**



Evaluating a buffer

You now have a file with three expressions in it, one of which is actually composed of two functions, `setf` and `get-string-from-user`. It should look like this:

```
(defvar your-name)

(setf your-name (get-string-from-user
            "Please type in your name."))

(format t "Hello, ~A!" your-name)
```

Now get the result of these expressions:

- **Evaluate the buffer.**

  Choose Eval Buffer from the Eval menu.

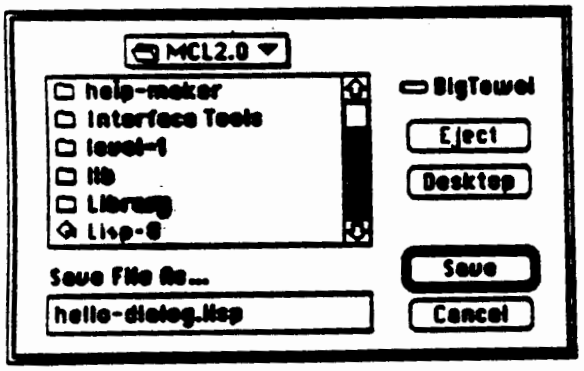
Saving a buffer to a file

To preserve code, you need to save it to a file.

1. **Choose Save from the File menu or press Command-S.**

   If you are working with a new file, Macintosh Common Lisp prompts you for a name, using the dialog box shown in Figure 2-7.

- **Figure 2-7** The Save File dialog box



2. **Type the name `hello-dialog.lisp`. To accept the name, either press Return or click the Save button.**

Moving around in a Fred window

The file in your Fred window should now look like this:

```
"Hello, world!"
(get-string-from-user
            "Please type in your name.")
```

By simple editing, you can turn this into a more interesting program that gets your name and returns "`Hello,` *your-name*`!`"

1. **Create a symbol to represent the value of your name.**

   To do this, edit `get-string-from-user`.

   Move the mouse cursor to just before the opening parenthesis of `get-string-from-user` and click once to position the insertion point there.

   Type exactly the following, followed by a space:

   ```
   (defvar your-name)

   (setf your-name
   ```

   Position the mouse cursor just beyond the closing parenthesis of the `get-string-from-user` expression and click again. Type a second closing parenthesis there.

   This is what you should see:

   ```
   (setf your-name (get-string-from-user
         "Please type in your name."))
   ```

   You are asking Macintosh Common Lisp to perform a `get-string-from-user`, which returns your name, then to set the symbol `your-name` to that value.

   The use of the `defvar` macro declares that the variable `your-name` is a special variable in this program.

2. **Evaluate the expression.**

   Double-click just after the second closing parenthesis and press Enter or Command-E. The dialog box will appear and, after you type your name, it will be printed in the Listener.

Evaluating in a Fred window

When you are working in a Fred window, you can evaluate expressions by positioning your insertion point just after the expression and pressing Enter (not the Return key, but the Enter key on the numeric keypad) or Command-E.

You can also make a selection and then evaluate it:

1. **Select the string `"Hello, world!"`.**

   The entire expression is highlighted.

2. **Evaluate the expression by pressing Command-E or Enter.**

The results of an evaluation always appear in the Listener, where you should see `"Hello, world!"`

Counting parentheses and highlighting an expression

Traditionally, one of the problems in programming Lisp is balancing parentheses. Fred shows you where an expression begins and ends by causing one of a pair of parentheses to blink when you place the insertion point by the other.

1. **Press Return and type the following code, which asks your name:**

   ```
   (get-string-from-user
         "Please type in your name.")
   ```

   The opening parenthesis of the expression blinks when the insertion point is just after the closing parenthesis.

   When you double-click just after one parenthesis of a pair, Fred selects everything between it and the other parenthesis.

2. **Position the insertion point just after the closing parenthesis and double-click.**

   The entire expression is highlighted. If this were an expression within an expression, only the part of the expression back to the corresponding open parenthesis would be highlighted.

   Now evaluate it.

3. **Press Command-E or Enter.**

   Macintosh Common Lisp evaluates the expression. Figure 2-6 shows the result.

Fred and the Listener work in tandem.

- You usually develop code in a Fred window (which is perhaps a copy of a Listener session).
- You see the results of evaluating code in the Listener.
- Almost all of the Fred editing features work the same way in Fred windows and in the Listener.

To get into Fred, simply open a file or create a new file.

In the next few sections, you will learn how to do the following in a Fred window:

- create a new window
- add text
- select text
- evaluate expressions
- copy text
- save work to a file on disk

You'll create a dialog box that asks your name and prints "`Hello,` *your-name*`!`"

Creating a Fred window

Fred windows are created whenever you open a new or an existing file:

- **Open a new file.**

  Choose New from the File menu.

  You see two windows, your original Listener and a new Fred window, as in Figure 2-5. (Their onscreen positions may be different from those in the figure.) The new Fred window becomes the active window.

This interaction shows that you can preserve information across Listener windows:

1. **Set a symbol to a value.**

```
? (setf x 5)
5
```

2. **Close the Listener window.**

   Click the close box in the upper-left corner of the Listener.

   The Listener window closes.

   Macintosh Common Lisp automatically creates a fresh Listener window.

3. **Evaluate the symbol x in the new Listener.**

   Type x in the new Listener.

```
? x
5
```

But the Listener doesn't preserve information from session to session; it simply reads code, evaluates it interactively, and prints a result.

If you set a symbol to a value in the Listener—for instance, if you give x the value 5—Macintosh Common Lisp preserves that information until the end of the current session. But, if you were to close Macintosh Common Lisp and open it again, x would no longer have a value:

```
? x
> Error: Unbound variable: X
> While executing: SYMBOL-VALUE
```

(You can cancel this error—"abort" it—by pressing Command-period or choosing Abort from the Eval menu.)

Preserving work from session to session

To preserve work from session to session, you must save it as source code. All source code in Lisp is simply text files, which you can edit with the Macintosh Common Lisp editor, Fred, described later in this chapter in "The MCL Editor, Fred."

You can save a copy of the Listener to a file that you can edit in Fred, using Save Copy As from the File menu. However, you will more often work directly from Fred and evaluate your code in the Listener.

You'll recognize the Listener from its name in the title bar, "Listener," and from the question mark prompt. Whenever you see this question mark, the Listener is ready to read input.

The Listener is natural and straightforward to use. Text you enter into the Listener appears in **boldface**. Macintosh Common Lisp's response appears in Normal type.

When you see examples in the text, the bold text indicates what you type. The plain text shows what Macintosh Common Lisp prints in response.

? **"Hello, world!"**	<—This is what you type.
"Hello, world!"	<—This is Macintosh Common Lisp's response.

Evaluating expressions in the Listener

When you type an expression and press Return, Macintosh Common Lisp immediately evaluates the expression—finds its value—and returns a result. You can see an example by getting the Listener to evaluate a string.

Among the simplest Lisp expressions are strings. A string is a one-dimensional array whose elements are characters. In Lisp, the beginning and end of a string are indicated by double quotation marks.

Here are some examples of strings:

```
"dog"
"cat"
"XJ313"
```

The value of a string in Lisp is the string itself:

```
? "dog"
"dog"
? "cat"
"cat"
? "XJ313"
"XJ313"
```

Here's a simple evaluation:

- Type **"Hello, world!"** and then press Return.

List Definitions Displays a modeless dialog box containing a table of all the definitions in the editor buffer displayed in the frontmost window. The user can select a definition, and the buffer will scroll to that definition.

This command is disabled when the active window is not an editor window.

Search Files Displays a dialog box for searching a set of files for a given string. The user can use wildcard characters to specify the set of files to search. (Macintosh Common Lisp supports Common Lisp extended wildcards, which are documented in *Common Lisp: The Language;* see also the section "Wildcards" in Chapter 9, "File System Interface," of the *Macintosh Common Lisp Reference.*)

Backtrace (⌘-B)

This command is enabled only when Macintosh Common Lisp is in a break loop. Choosing it brings up the Stack Backtrace window, described in Chapter 4, "Debugging."

Fred Commands Displays a window listing all the commands available in Fred, the Macintosh Common Lisp editor. You can search through this list using the Fred search command, Control-S, and can copy text from it to another Fred window. See the section "Getting Help on Fred Commands," later in this chapter.

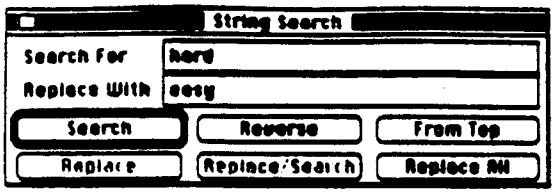
Listener Commands

Brings up a window listing all Listener commands that differ from Fred commands.

Print Options... Brings up a dialog box that allows the user to set the values of several Lisp printer parameters. These are described in Chapter 1, "The Macintosh Common Lisp Environment," in the *Macintosh Common Lisp Reference.*

Environment... Brings up a dialog box that allows the user to set the values of several global variables affecting the Macintosh Common Lisp environment. These are described in Chapter 1, "The Macintosh Common Lisp Environment," in the *Macintosh Common Lisp Reference.*

■ **Figure 2-2** The String Search dialog box



Eval menu commands

The Eval menu contains the following eight commands:

Eval Selection (⌘-E)

Evaluates the current selection in the top editor buffer. (The buffer is contained in the window; it is a copy of the file being edited.) If there is no selection and the insertion point is next to a parenthesis, the expression bounded by the parentheses is evaluated.

Eval Buffer (⌘-H)

Evaluates the entire contents of the top editor buffer. This is equivalent to choosing Select All followed by Eval Selection.

Load... (⌘-Y) Allows you to select a file for loading into Macintosh Common Lisp. You may load both `lisp` and `fasl` files (respectively, uncompiled and compiled Lisp files).

Compile File... Allows you to select a file for compilation. You are asked to specify both the source and destination files.

Abort (⌘-period)

Cancels the current computation and returns to the read-eval-print loop (the functionality through which Macintosh Common Lisp reads an expression, evaluates it, and prints the result). If Macintosh Common Lisp was in a break loop, it leaves the loop. A canceled computation cannot be resumed. You'll learn more about this command in the section "Recovering From an Error With Abort," in Chapter 4.

m-U	ED-UPCASE-WORD
c-v	ED-NEXT-SCREEN
c-shift-V	ED-SELECT-NEXT-SCREEN
m-v	ED-PREVIOUS-SCREEN
m-shift-V	ED-SELECT-PREVIOUS-SCREEN
c-w	ED-KILL-REGION
m-w	ED-COPY-REGION-AS-KILL
c-x c-Space	ED-DELETE-FORWARD-WHITESPACE
c-x ;	ED-SET-COMMENT-COLUMN
c-x c-a	ED-ARGLIST
c-x c-c	ED-EVAL-OR-COMPILE-TOP-LEVEL-SEXP
c-x c-d	ED-GET-DOCUMENTATION
c-x c-e	ED-EVAL-CURRENT-SEXP
c-x h	SELECT-ALL
c-x c-i	ED-INSPECT-CURRENT-SEXP
c-x c-m	ED-MACROEXPAND-CURRENT-SEXP
c-x c-r	ED-READ-CURRENT-SEXP
c-x c-s	WINDOW-SAVE
c-x c-u	ED-HISTORY-UNDO
c-x c-v	EDIT-SELECT-FILE
c-x c-w	WINDOW-SAVE-AS
c-x c-x	ED-EXCHANGE-POINT-AND-MARK
c-y	ED-YANK
m-y	ED-YANK-POP
m-\	ED-DELETE-HORIZONTAL-WHITESPACE
c-_	ED-HISTORY-UNDO
c-m-_	ED-PRINT-HISTORY
m-~	WINDOW-SET-NOT-MODIFIED
DEL	ED-DELETE-CHAR
0	ED-DIGIT
m-0	ED-NUMERIC-ARGUMENT
c-0	ED-NUMERIC-ARGUMENT
c-m-0	ED-NUMERIC-ARGUMENT
1	ED-DIGIT
m-1	ED-NUMERIC-ARGUMENT
c-1	ED-NUMERIC-ARGUMENT
c-m-1	ED-NUMERIC-ARGUMENT
2	ED-DIGIT
m-2	ED-NUMERIC-ARGUMENT
c-2	ED-NUMERIC-ARGUMENT
c-m-2	ED-NUMERIC-ARGUMENT
3	ED-DIGIT
m-3	ED-NUMERIC-ARGUMENT
c-3	ED-NUMERIC-ARGUMENT
c-m-3	ED-NUMERIC-ARGUMENT
4	ED-DIGIT
m-4	ED-NUMERIC-ARGUMENT
c-4	ED-NUMERIC-ARGUMENT
c-m-4	ED-NUMERIC-ARGUMENT
5	ED-DIGIT
m-5	ED-NUMERIC-ARGUMENT
c-5	ED-NUMERIC-ARGUMENT
c-m-5	ED-NUMERIC-ARGUMENT
6	ED-DIGIT
m-6	ED-NUMERIC-ARGUMENT
c-6	ED-NUMERIC-ARGUMENT
c-m-6	ED-NUMERIC-ARGUMENT
7	ED-DIGIT
m-7	ED-NUMERIC-ARGUMENT
c-7	ED-NUMERIC-ARGUMENT
c-m-7	ED-NUMERIC-ARGUMENT
8	ED-DIGIT
m-8	ED-NUMERIC-ARGUMENT

c-8	ED-NUMERIC-ARGUMENT
c-m-8	ED-NUMERIC-ARGUMENT
9	ED-DIGIT
m-9	ED-NUMERIC-ARGUMENT
c-9	ED-NUMERIC-ARGUMENT
c-m-9	ED-NUMERIC-ARGUMENT

Listener commands that differ from Fred Commands:
c = control [control key or command + shift keys]
m = meta [option key]

Enter	ED-COPY-ENTER-COMMAND
Return	ED-ENTER-COMMAND
c-Return	ED-SELF-INSERT
m-Return	ED-COPY-ENTER-COMMAND
c-?	ED-LISTENER-HELP
m-g	ED-GRAB-LAST-INPUT
c-g	ED-ABORT-LISTENER-INPUT
c-m-n	ED-NEXT-INPUT-LINE
c-m-p	ED-LAST-INPUT-LINE

Global command table:
c = control [control key or command + shift keys]
m = meta [option key]

Home	ED-BEGINNING-OF-BUFFER
Enter	ED-EVAL-OR-COMPILE-CURRENT-SEXP
End	ED-END-OF-BUFFER
Help	ED-INSPECT-CURRENT-SEXP
Delete	ED-RUBOUT-CHAR
shift-Delete	ED-RUBOUT-CHAR
m-Delete	ED-RUBOUT-WORD
c-m-Delete	ED-KILL-BACKWARD-SEXP
Tab	ED-INDENT-FOR-LISP
c-Tab	ED-INDENT-DIFFERENTLY
m-Tab	ED-SELF-INSERT
PageUp	ED-PREVIOUS-SCREEN
shift-PageUp	ED-SELECT-PREVIOUS-SCREEN
Page	ED-NEXT-SCREEN
shift-Page	ED-SELECT-NEXT-SCREEN
Return	ED-SELF-INSERT
c-Return	ED-NEWLINE-AND-INDENT
ESC	ED-COLLAPSE-SELECTION
BackArrow	ED-BACKWARD-CHAR
m-BackArrow	ED-BACKWARD-WORD
shift-BackArrow	ED-BACKWARD-SELECT-CHAR
m-shift-BackArrow	ED-BACKWARD-SELECT-WORD
c-BackArrow	ED-BACKWARD-SEXP
c-shift-BackArrow	ED-BACKWARD-SELECT-SEXP
c-m-BackArrow	ED-BACKWARD-SEXP
c-m-shift-BackArrow	ED-BACKWARD-SELECT-SEXP
ForwardArrow	ED-FORWARD-CHAR
m-ForwardArrow	ED-FORWARD-WORD
shift-ForwardArrow	ED-FORWARD-SELECT-CHAR
m-shift-ForwardArrow	ED-FORWARD-SELECT-WORD
c-ForwardArrow	ED-FORWARD-SEXP
c-shift-ForwardArrow	ED-FORWARD-SELECT-SEXP
c-m-ForwardArrow	ED-FORWARD-SEXP
c-m-shift-ForwardArrow	ED-FORWARD-SELECT-SEXP
UpArrow	ED-PREVIOUS-LINE
shift-UpArrow	ED-SELECT-PREVIOUS-LINE
c-m-UpArrow	ED-PREVIOUS-LIST
c-m-shift-UpArrow	ED-SELECT-PREVIOUS-LIST
DownArrow	ED-NEXT-LINE
shift-DownArrow	ED-SELECT-NEXT-LINE
c-m-DownArrow	ED-NEXT-LIST
c-m-shift-DownArrow	ED-SELECT-NEXT-LIST
Space	INTERACTIVE-ARGLIST
m-Space	ED-DELETE-WHITESPACE
c-m-Space	ED-SELECT-CURRENT-SEXP
c-Space	ED-PUSH/POP-MARK-RING
m-"	ED-INSERT-DOUBLE-QUOTES
m-#	ED-INSERT-SHARP-COMMENT
m-(	ED-INSERT-PARENS
c-m-(	ED-BWD-UP-LIST
m-)	ED-MOVE-OVER-CLOSE-AND-REINDENT
c-m-)	ED-FWD-UP-LIST
c-+	ED-WHAT-CURSOR-POSITION
-	ED-DIGIT
m--	ED-NUMERIC-ARGUMENT
c--	ED-NUMERIC-ARGUMENT
c-m--	ED-NUMERIC-ARGUMENT
m-.	ED-EDIT-DEFINITION
c-/	ED-HELP

f-1	UNDO
f-2	CUT
f-3	COPY
f-4	PASTE
m-;	ED-INDENT-COMMENT
c-m-;	ED-KILL-COMMENT
m-<	ED-BEGINNING-OF-BUFFER
c-=	ED-WHAT-CURSOR-POSITION
m->	ED-END-OF-BUFFER
c-?	ED-HELP
c-m-@	Same as c-m-Space
c-@	Same as c-Space
c-a	ED-BEGINNING-OF-LINE
c-shift-A	ED-SELECT-BEGINNING-OF-LINE
c-m-a	ED-START-TOP-LEVEL-SEXP
c-m-A	ED-START-TOP-LEVEL-SEXP
c-b	ED-BACKWARD-CHAR
c-shift-B	ED-BACKWARD-SELECT-CHAR
m-b	ED-BACKWARD-WORD
m-shift-B	ED-BACKWARD-SELECT-WORD
c-m-b	ED-BACKWARD-SEXP
c-m-shift-B	ED-BACKWARD-SELECT-SEXP
m-c	ED-CAPITALIZE-WORD
c-d	ED-DELETE-CHAR
m-d	ED-DELETE-WORD
c-m-d	ED-DOWN-LIST
c-e	ED-END-OF-LINE
c-shift-E	ED-SELECT-END-OF-LINE
c-m-e	ED-END-TOP-LEVEL-SEXP
c-m-E	ED-END-TOP-LEVEL-SEXP
c-f	ED-FORWARD-CHAR
c-shift-F	ED-FORWARD-SELECT-CHAR
m-f	ED-FORWARD-WORD
m-shift-F	ED-FORWARD-SELECT-WORD
c-m-f	ED-FORWARD-SEXP
c-m-shift-F	ED-FORWARD-SELECT-SEXP
c-m-h	ED-SELECT-TOP-LEVEL-SEXP
c-k	ED-KILL-LINE
c-m-k	ED-KILL-FORWARD-SEXP
m-l	ED-DOWNCASE-WORD
c-m-l	ED-LAST-BUFFER
c-m	ED-MACROEXPAND-1-CURRENT-SEXP
m-m	ED-BACK-TO-INDENTATION
c-m-m	REPARSE-MODELINE
c-m-M	ADD-MODELINE
c-n	ED-NEXT-LINE
c-N	ED-SELECT-NEXT-LINE
c-m-n	ED-NEXT-LIST
c-m-shift-N	ED-SELECT-NEXT-LIST
c-o	ED-OPEN-LINE
c-m-o	ED-SPLIT-LINE
c-p	ED-PREVIOUS-LINE
c-shift-P	ED-SELECT-PREVIOUS-LINE
c-m-p	ED-PREVIOUS-LIST
c-m-shift-P	ED-SELECT-PREVIOUS-LIST
c-m-q	ED-INDENT-SEXP
c-r	ED-I-SEARCH-REVERSE
c-s	ED-I-SEARCH-FORWARD
c-t	ED-TRANSPOSE-CHARS
m-t	ED-TRANSPOSE-WORDS
c-m-t	ED-TRANSPOSE-SEXPS
c-u	ED-UNIVERSAL-ARGUMENT
c-m-u	ED-BWD-UP-LIST

PWA CUMULATIVE REPORT

CUMULATIVE MATERIALS PROCESSED

LATHE-LOCATION LATHE-WC-3 Jan 1, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

LATHE-LOCATION LATHE-WC-3 Jan 2, 1993 08:00

Batches Started--3 Batches Complete--2 Hours Run-- 23.75

MACHINE-OIL	10.0000000
STEEL	200.0000000

LATHE-LOCATION LATHE-WC-3 Jan 3, 1993 08:00

Batches Started--3 Batches Complete--3 Hours Run-- 24.00

MACHINE-OIL	15.0000000
STEEL	300.0000000

LATHE-LOCATION LATHE-WC-3 Jan 4, 1993 07:00

Batches Started--3 Batches Complete--3 Hours Run-- 24.00

MACHINE-OIL	15.0000000
STEEL	300.0000000

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 1, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 2, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 3, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 4, 1993 07:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 1, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 2, 1993 08:00

Batches Started--1 Batches Complete--1 Hours Run-- 2.00

KIM-WIPES	5.0000000
MACHINE-OIL	10.0000000
SOLVENT	24.0000000
STEEL	2.0000000

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 3, 1993 08:00

Batches Started--1 Batches Complete--1 Hours Run-- 2.00

KIM-WIPES	5.0000000
MACHINE-OIL	10.0000000
SOLVENT	24.0000000
STEEL	2.0000000

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 4, 1993 07:00

Batches Started--1 Batches Complete--1 Hours Run-- 2.00

KIM-WIPES	5.0000000
MACHINE-OIL	10.0000000
SOLVENT	24.0000000
STEEL	2.0000000

PWA FEED REPORT

CUMULATIVE MATERIAL RECEIPTS

LATHE-LOCATION Jan 1, 1993 08:00

LATHE-LOCATION Jan 2, 1993 08:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

LATHE-LOCATION Jan 3, 1993 08:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

LATHE-LOCATION Jan 4, 1993 07:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

RECEIVING Jan 1, 1993 08:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

RECEIVING Jan 2, 1993 08:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

RECEIVING Jan 3, 1993 08:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

RECEIVING Jan 4, 1993 07:00
From PARTS
Source RAW-PARTS
STEEL 300.0000000

SHIPPING-LOCATION Jan 1, 1993 08:00

SHIPPING-LOCATION Jan 2, 1993 08:00
From PARTS
Source LATHED-PARTS
STEEL 198.0000000

SHIPPING-LOCATION Jan 3, 1993 08:00
From PARTS
Source LATHED-PARTS
STEEL 297.0000000

SHIPPING-LOCATION Jan 4, 1993 07:00
From PARTS
Source LATHED-PARTS
STEEL 297.0000000

WASTE-STORAGE-LOCATION Jan 1, 1993 08:00
From PARTS

WASTE-STORAGE-LOCATION Jan 2, 1993 08:00
From PARTS
Source BAG-OF-WASTE

```
         KIM-WIPES                                  5.0000000
         MACHINE-OIL                               10.0000000
         SOLVENT                                   24.0000000
   From PARTS

WASTE-STORAGE-LOCATION Jan  3, 1993 08:00
   From PARTS
      Source BAG-OF-WASTE
         KIM-WIPES                                  5.0000000
         MACHINE-OIL                               10.0000000
         SOLVENT                                   24.0000000
   From PARTS

WASTE-STORAGE-LOCATION Jan  4, 1993 07:00
   From PARTS
      Source BAG-OF-WASTE
         KIM-WIPES                                  5.0000000
         MACHINE-OIL                               10.0000000
         SOLVENT                                   24.0000000
   From PARTS

WASTE-TREATMENT-LOCATION Jan  1, 1993 08:00
   From PARTS

WASTE-TREATMENT-LOCATION Jan  2, 1993 08:00
   From WASTE-TREATMENT-WC-3
      Source CLEANED-STEEL-VAT
         STEEL                                      2.0000000
   From PARTS
   From LATHE-WC-3
      Source WASTE-TREATMENT-VAT
         MACHINE-OIL                               10.0000000
         STEEL                                      2.0000000
   From PARTS

WASTE-TREATMENT-LOCATION Jan  3, 1993 08:00
   From WASTE-TREATMENT-WC-3
      Source CLEANED-STEEL-VAT
         STEEL                                      2.0000000
   From PARTS
   From LATHE-WC-3
      Source WASTE-TREATMENT-VAT
         MACHINE-OIL                               15.0000000
         STEEL                                      3.0000000
   From PARTS

WASTE-TREATMENT-LOCATION Jan  4, 1993 07:00
   From WASTE-TREATMENT-WC-3
      Source CLEANED-STEEL-VAT
         STEEL                                      2.0000000
   From PARTS
   From LATHE-WC-3
      Source WASTE-TREATMENT-VAT
         MACHINE-OIL                               15.0000000
         STEEL                                      3.0000000
   From PARTS
```

PWA PROCESSING INCREMENTAL REPORT

INCREMENTAL MATERIALS PROCESSED

LATHE-LOCATION LATHE-WC-3 Jan 1, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

LATHE-LOCATION LATHE-WC-3 Jan 2, 1993 08:00

Batches Started--3 Batches Complete--2 Hours Run-- 23.75

MACHINE-OIL	10.0000000
STEEL	200.0000000

LATHE-LOCATION LATHE-WC-3 Jan 3, 1993 08:00

Batches Started--0 Batches Complete--1 Hours Run-- 0.25

MACHINE-OIL	5.0000000
STEEL	100.0000000

LATHE-LOCATION LATHE-WC-3 Jan 4, 1993 07:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

MACHINE-OIL	0.0000000
STEEL	0.0000000

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 1, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 2, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 3, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-2 Jan 4, 1993 07:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 1, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 2, 1993 08:00

Batches Started--1 Batches Complete--1 Hours Run-- 2.00

KIM-WIPES	5.0000000
MACHINE-OIL	10.0000000
SOLVENT	24.0000000
STEEL	2.0000000

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 3, 1993 08:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

KIM-WIPES	0.0000000
MACHINE-OIL	0.0000000
SOLVENT	0.0000000
STEEL	0.0000000

WASTE-TREATMENT-LOCATION WASTE-TREATMENT-WC-3 Jan 4, 1993 07:00

Batches Started--0 Batches Complete--0 Hours Run-- 0.00

KIM-WIPES	0.0000000
MACHINE-OIL	0.0000000
SOLVENT	0.0000000
STEEL	0.0000000

PWA MATERIAL-HANDLING REPORT

MATERIALS HANDLED — Simulation Time: 1/1/1993 8:0

Process Name: LATHE-WC-3 — Location: LATHE-LOCATION

Hours Run: 0.00 — Number of Starts: 0

Percent Utilization: UNDEFINED % — Number Completions: 0

Based On: NOT-SPECIFIED

MATERIALS HANDLED — Simulation Time: 1/2/1993 8:0

Process Name: LATHE-WC-3 — Location: LATHE-LOCATION

Hours Run: 23.75 — Number of Starts: 3

Percent Utilization: UNDEFINED % — Number Completions: 2

Based On: NOT-SPECIFIED

Material Name	Material Receipts	Mat Added	Qty Process	Fee Sto
MACHINE-OIL	0.0000000	10.0000000	10.0000000	0.
STEEL	300.0000000	0.0000000	200.0000000	99.

OUTPUTS

	LATHED-PARTS	WASTE-TREATMENT-LOCATION
MACHINE-OIL	0.0000000	10.0000000
STEEL	198.0000000	2.0000000

MATERIALS HANDLED Simulation Time: 1/3/1993 8:0

Process Name: LATHE-WC-3 Location: LATHE-LOCATION

Hours Run: 24.00 Number of Starts: 3

Percent Utilization: UNDEFINED % Number Completions: 3

Based On: NOT-SPECIFIED

Material Name	Material Receipts	Mat Added	Qty Process	Fee Sto
MACHINE-OIL	0.0000000	15.0000000	15.0000000	0.
STEEL	300.0000000	0.0000000	300.0000000	0.

OUTPUTS

	LATHED-PARTS	WASTE-TREATMENT-LOCATION
MACHINE-OIL	0.0000000	15.0000000
STEEL	297.0000000	3.0000000

MATERIALS HANDLED Simulation Time: 1/4/1993 7:0

Process Name: LATHE-WC-3 Location: LATHE-LOCATION

Hours Run: 24.00 Number of Starts: 3

Percent Utilization: UNDEFINED % Number Completions: 3

Based On: NOT-SPECIFIED

Material	Material	Mat	Qty	Fee

Name	Receipts	Added	Process	Sto ↲ re
MACHINE-OIL	0.0000000	15.0000000	15.0000000	0. ↲ 0000000
STEEL	300.0000000	0.0000000	300.0000000	0. ↲ 0000000

OUTPUTS

	LATHED-PARTS	WASTE-TREATMENT-LOCATION
MACHINE-OIL	0.0000000	15.0000000
STEEL	297.0000000	3.0000000

MATERIALS HANDLED Simulation Time: 1/1/1993 8:0

Location: RECEIVING

Material Name	Material Receipts	Feed Store
STEEL	300.0000000	0.0000000

MATERIALS HANDLED Simulation Time: 1/2/1993 8:0

Location: RECEIVING

Material Name	Material Receipts	Feed Store
STEEL	300.0000000	0.0000000

MATERIALS HANDLED Simulation Time: 1/3/1993 8:0

Location: RECEIVING

Material Name	Material Receipts	Feed Store
STEEL	300.0000000	0.0000000

MATERIALS HANDLED Simulation Time: 1/4/1993 7:0

Location: RECEIVING

Material Name	Material Receipts	Feed Store

STEEL	300.0000000	0.0000000

MATERIALS HANDLED Simulation Time: 1/1/1993 8:0

Location: SHIPPING-LOCATION

MATERIALS HANDLED Simulation Time: 1/2/1993 8:0

Location: SHIPPING-LOCATION

Material Name	Material Receipts	Feed Store
STEEL	198.0000000	198.0000000

MATERIALS HANDLED Simulation Time: 1/3/1993 8:0

Location: SHIPPING-LOCATION

Material Name	Material Receipts	Feed Store
STEEL	297.0000000	297.0000000

MATERIALS HANDLED Simulation Time: 1/4/1993 7:0

Location: SHIPPING-LOCATION

Material Name	Material Receipts	Feed Store
STEEL	297.0000000	297.0000000

MATERIALS HANDLED Simulation Time: 1/1/1993 8:0

Location: WASTE-STORAGE-LOCATION

MATERIALS HANDLED Simulation Time: 1/2/1993 8:0

Location: WASTE-STORAGE-LOCATION

Material Name	Material Receipts	Feed Store
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	10.0000000	10.0000000
SOLVENT	24.0000000	24.0000000

MATERIALS HANDLED Simulation Time: 1/3/1993 8:0

Location: WASTE-STORAGE-LOCATION

Material Name	Material Receipts	Feed Store
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	10.0000000	10.0000000
SOLVENT	24.0000000	24.0000000

MATERIALS HANDLED Simulation Time: 1/4/1993 7:0

Location: WASTE-STORAGE-LOCATION

Material Name	Material Receipts	Feed Store
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	10.0000000	10.0000000
SOLVENT	24.0000000	24.0000000

MATERIALS HANDLED Simulation Time: 1/1/1993 8:0

Location: WASTE-TREATMENT-LOCATION

Process Name: WASTE-TREATMENT-WC-2

Hours Run: 0.00 Number of Starts: 0

Percent Utilization: UNDEFINED % Number Completions: 0

Based On: NOT-SPECIFIED

Process Name: WASTE-TREATMENT-WC-3

Hours Run: 0.00 Number of Starts: 0

Percent Utilization: UNDEFINED % Number Completions: 0

Based On: NOT-SPECIFIED

MATERIALS HANDLED Simulation Time: 1/2/1993 8:0

Location: WASTE-TREATMENT-LOCATION

Material Name	Material Receipts	Feed Store
MACHINE-OIL	10.0000000	0.0000000
STEEL	4.0000000	2.0000000

Process Name: WASTE-TREATMENT-WC-2

Hours Run: 0.00 Number of Starts: 0

Percent Utilization: UNDEFINED % Number Completions: 0

Based On: NOT-SPECIFIED

Process Name: WASTE-TREATMENT-WC-3

Hours Run: 2.00 Number of Starts: 1

Percent Utilization: UNDEFINED % Number Completions: 1

Based On: NOT-SPECIFIED

Material Name	Mat Added	Qty Process
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	0.0000000	10.0000000
SOLVENT	24.0000000	24.0000000
STEEL	0.0000000	2.0000000

OUTPUTS

	BAG-OF-WASTE	WASTE-TREATMENT-LOCATION
KIM-WIPES	5.0000000	0.0000000
MACHINE-OIL	10.0000000	0.0000000
SOLVENT	24.0000000	0.0000000
STEEL	0.0000000	2.0000000

MATERIALS HANDLED Simulation Time: 1/3/1993 8:0

Location: WASTE-TREATMENT-LOCATION

Material Name	Material Receipts	Feed Store
MACHINE-OIL	15.0000000	5.0000000
STEEL	5.0000000	3.0000000

Process Name: WASTE-TREATMENT-WC-2

Hours Run: 0.00 Number of Starts: 0

Percent Utilization: UNDEFINED % Number Completions: 0

Based On: NOT-SPECIFIED

Process Name: WASTE-TREATMENT-WC-3

Hours Run: 2.00 Number of Starts: 1

Percent Utilization: UNDEFINED % Number Completions: 1

Based On: NOT-SPECIFIED

Material Name	Mat Added	Qty Process
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	0.0000000	10.0000000
SOLVENT	24.0000000	24.0000000
STEEL	0.0000000	2.0000000

OUTPUTS

	BAG-OF-WASTE	WASTE-TREATMENT-LOCATION
KIM-WIPES	5.0000000	0.0000000
MACHINE-OIL	10.0000000	0.0000000
SOLVENT	24.0000000	0.0000000
STEEL	0.0000000	2.0000000

MATERIALS HANDLED Simulation Time: 1/4/1993 7:0

Location: WASTE-TREATMENT-LOCATION

Material Name	Material Receipts	Feed Store
MACHINE-OIL	15.0000000	5.0000000
STEEL	5.0000000	3.0000000

Process Name: WASTE-TREATMENT-WC-2

Hours Run: 0.00 Number of Starts: 0

Percent Utilization: UNDEFINED % Number Completions: 0

Based On: NOT-SPECIFIED

Process Name: WASTE-TREATMENT-WC-3

Hours Run: 2.00 Number of Starts: 1

Percent Utilization: UNDEFINED % Number Completions: 1

Based On: NOT-SPECIFIED

Material Name	Mat Added	Qty Process
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	0.0000000	10.0000000
SOLVENT	24.0000000	24.0000000
STEEL	0.0000000	2.0000000

OUTPUTS

	BAG-OF-WASTE	WASTE-TREATMENT-LOCATION
KIM-WIPES	5.0000000	0.0000000
MACHINE-OIL	10.0000000	0.0000000
SOLVENT	24.0000000	0.0000000
STEEL	0.0000000	2.0000000

PWA MATERIALS REPORT

CUMULATIVE MATERIAL SUMMARY

--- LATHE-LOCATION

	Received	Remaining
Jan 1, 1993 08:00		
Jan 2, 1993 08:00		
STEEL	300.0000000	99.0000000
Jan 3, 1993 08:00		
STEEL	300.0000000	0.0000000
Jan 4, 1993 07:00		
STEEL	300.0000000	0.0000000

--- RECEIVING

	Received	Remaining
Jan 1, 1993 08:00		
STEEL	300.0000000	0.0000000
Jan 2, 1993 08:00		
STEEL	300.0000000	0.0000000
Jan 3, 1993 08:00		
STEEL	300.0000000	0.0000000
Jan 4, 1993 07:00		
STEEL	300.0000000	0.0000000

--- SHIPPING-LOCATION

	Received	Remaining
Jan 1, 1993 08:00		
Jan 2, 1993 08:00		
STEEL	198.0000000	198.0000000
Jan 3, 1993 08:00		
STEEL	297.0000000	297.0000000
Jan 4, 1993 07:00		
STEEL	297.0000000	297.0000000

--- WASTE-STORAGE-LOCATION

	Received	Remaining
Jan 1, 1993 08:00		
Jan 2, 1993 08:00		
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	10.0000000	10.0000000
SOLVENT	24.0000000	24.0000000
Jan 3, 1993 08:00		
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	10.0000000	10.0000000
SOLVENT	24.0000000	24.0000000
Jan 4, 1993 07:00		
KIM-WIPES	5.0000000	5.0000000
MACHINE-OIL	10.0000000	10.0000000
SOLVENT	24.0000000	24.0000000

--- WASTE-TREATMENT-LOCATION

	Received	Remaining
Jan 1, 1993 08:00		
Jan 2, 1993 08:00		
MACHINE-OIL	10.0000000	0.0000000
STEEL	4.0000000	2.0000000
Jan 3, 1993 08:00		
MACHINE-OIL	15.0000000	5.0000000
STEEL	5.0000000	3.0000000
Jan 4, 1993 07:00		
MACHINE-OIL	15.0000000	5.0000000
STEEL	5.0000000	3.0000000

PWA PART-HANDLING REPORT

PART HOLDINGS

--- LATHE-LOCATION

Process	Parts In	Total Remain Parts	Stored	Done	Pending Transfer	In P
Jan 1, 1993 08:00						
Jan 2, 1993 08:00						
LATHED-PARTS	3.	1.	0.	0.	0.	1
RAW-PARTS	3.	0.	0.	0.	0.	0
Jan 3, 1993 08:00						
LATHED-PARTS	0.	0.	0.	0.	0.	0
RAW-PARTS	0.	0.	0.	0.	0.	0
Jan 4, 1993 07:00						
LATHED-PARTS	0.	0.	0.	0.	0.	0
RAW-PARTS	0.	0.	0.	0.	0.	0

--- RECEIVING

Process	Parts In	Total Remain Parts	Stored	Done	Pending Transfer	In P
Jan 1, 1993 08:00						
RAW-PARTS	3.	0.	0.	0.	0.	0
Jan 2, 1993 08:00						
RAW-PARTS	0.	0.	0.	0.	0.	0
Jan 3, 1993 08:00						

RAW-PARTS	0.	0.	0.	0.	0.	0

Jan 4, 1993 07:00

RAW-PARTS	0.	0.	0.	0.	0.	0

--- SHIPPING-LOCATION

Process	Parts In	Total Remain Parts	Stored	Done	Pending Transfer	In P
Jan 1, 1993 08:00						
Jan 2, 1993 08:00						
LATHED-PARTS	2.	2.	2.	0.	0.	0
Jan 3, 1993 08:00						
LATHED-PARTS	1.	3.	3.	0.	0.	0
Jan 4, 1993 07:00						
LATHED-PARTS	0.	3.	3.	0.	0.	0

--- WASTE-STORAGE-LOCATION

Process	Parts In	Total Remain Parts	Stored	Done	Pending Transfer	In P
Jan 1, 1993 08:00						
Jan 2, 1993 08:00						
BAG-OF-WASTE	1.	1.	1.	0.	0.	0
Jan 3, 1993 08:00						
BAG-OF-WASTE	0.	1.	1.	0.	0.	0
Jan 4, 1993 07:00						
BAG-OF-WASTE	0.	1.	1.	0.	0.	0

--- WASTE-TREATMENT-LOCATION

Process	Parts In	Total Remain Parts	Stored	Done	Pending Transfer	In P
Jan 1, 1993 08:00						
Jan 2, 1993 08:00						
BAG-OF-WASTE	1.	0.	0.	0.	0.	0
Jan 3, 1993 08:00						
BAG-OF-WASTE	0.	0.	0.	0.	0.	0
Jan 4, 1993 07:00						
BAG-OF-WASTE	0.	0.	0.	0.	0.	0

PWA RCRA REPORT

POSSIBLE RCRA REGULATED PROCESS OUTPUTS

The following streams contain materials that may be subject to RCRA regulation. If you are uncertain if RCRA applies, contact EM Division for assistance.

There may be additional streams that are subject to RCRA which are not identified here.

RCRA Report by Material

Material	Class	Outfall	Location	Amount	Units
ARSENIC	RCRA	TEST-SAMPLES	TA-55	10.00	POUNDS

PWA MATERIAL-CLASS REPORT

Material-Class		Material	Outfall	Location
Amount	Units			
RCRA		ARSENIC	TEST-SAMPLES	TA-55
10.00	POUNDS			
TRU		PU	PU-SAMPLES	CMR-5117-VAULT
75.00	GRAMS			
		PU	PU-SAMPLES	CMR-5117
25.00	GRAMS			

PWA PROCESSING REPORT

CUMULATIVE MATERIALS PROCESSED

CMR-5117-VAULT TRAIN-1-494 Jan 3, 1992 08:00
Batches Started--0 Batches Complete--0 Hours Run-- 0.00

CMR-5117-VAULT TRAIN-1-494 Feb 1, 1992 08:00
Batches Started--0 Batches Complete--0 Hours Run-- 0.00

CMR-5117-VAULT TRAIN-1-494 Mar 1, 1992 08:00
Batches Started--0 Batches Complete--0 Hours Run-- 0.00

CMR-5117 TRAIN-1-495 Jan 3, 1992 08:00
Batches Started--0 Batches Complete--0 Hours Run-- 0.00

CMR-5117 TRAIN-1-495 Feb 1, 1992 08:00
Batches Started--0 Batches Complete--0 Hours Run-- 0.00

CMR-5117 TRAIN-1-495 Mar 1, 1992 08:00
Batches Started--0 Batches Complete--0 Hours Run-- 0.00

PWA RCRA-FIRST REPORTS

POSSIBLE RCRA REGULATED PROCESS OUTPUTS

The following streams contain materials that may be subject to RCRA regulation. If you are uncertain if RCRA applies, contact EM Division for assistance.

There may be additional streams that are subject to RCRA which are not identified here.

Outfall	Location	Material	Class	Amount
TEST-SAMPLES	TA-55	ARSENIC	RCRA	10.00

PWA ASCII REPORT

LATHE-WC-3-MACHINE-OIL-RECEIPTS
1/2/1993 8:00 0.0000000
1/3/1993 8:00 0.0000000
1/4/1993 7:00 0.0000000

LATHE-WC-3-MACHINE-OIL-MATS-ADDED
1/2/1993 8:00 10.0000000
1/3/1993 8:00 15.0000000
1/4/1993 7:00 15.0000000

LATHE-WC-3-MACHINE-OIL-QTY-PROCESSED
1/2/1993 8:00 10.0000000
1/3/1993 8:00 15.0000000
1/4/1993 7:00 15.0000000

LATHE-WC-3-MACHINE-OIL-FEED-STORE
1/2/1993 8:00 0.0000000
1/3/1993 8:00 0.0000000
1/4/1993 7:00 0.0000000

LATHE-WC-3-STEEL-RECEIPTS
1/2/1993 8:00 300.0000000
1/3/1993 8:00 300.0000000
1/4/1993 7:00 300.0000000

LATHE-WC-3-STEEL-MATS-ADDED
1/2/1993 8:00 0.0000000
1/3/1993 8:00 0.0000000
1/4/1993 7:00 0.0000000

LATHE-WC-3-STEEL-QTY-PROCESSED
1/2/1993 8:00 200.0000000
1/3/1993 8:00 300.0000000
1/4/1993 7:00 300.0000000

LATHE-WC-3-STEEL-FEED-STORE
1/2/1993 8:00 99.0000000
1/3/1993 8:00 0.0000000
1/4/1993 7:00 0.0000000

WASTE-TREATMENT-WC-3-KIM-WIPES-RECEIPTS
1/2/1993 8:00 0.0000000
1/3/1993 8:00 0.0000000
1/4/1993 7:00 0.0000000

WASTE-TREATMENT-WC-3-KIM-WIPES-MATS-ADDED
1/2/1993 8:00 5.0000000
1/3/1993 8:00 5.0000000
1/4/1993 7:00 5.0000000

WASTE-TREATMENT-WC-3-KIM-WIPES-QTY-PROCESSED
1/2/1993 8:00 5.0000000
1/3/1993 8:00 5.0000000
1/4/1993 7:00 5.0000000

WASTE-TREATMENT-WC-3-KIM-WIPES-FEED-STORE
1/2/1993 8:00 0.0000000
1/3/1993 8:00 0.0000000
1/4/1993 7:00 0.0000000

WASTE-TREATMENT-WC-3-MACHINE-OIL-RECEIPTS

1/2/1993 8:00	10.0000000
1/3/1993 8:00	15.0000000
1/4/1993 7:00	15.0000000

WASTE-TREATMENT-WC-3-MACHINE-OIL-MATS-ADDED

1/2/1993 8:00	0.0000000
1/3/1993 8:00	0.0000000
1/4/1993 7:00	0.0000000

WASTE-TREATMENT-WC-3-MACHINE-OIL-QTY-PROCESSED

1/2/1993 8:00	10.0000000
1/3/1993 8:00	10.0000000
1/4/1993 7:00	10.0000000

WASTE-TREATMENT-WC-3-MACHINE-OIL-FEED-STORE

1/2/1993 8:00	0.0000000
1/3/1993 8:00	5.0000000
1/4/1993 7:00	5.0000000

WASTE-TREATMENT-WC-3-SOLVENT-RECEIPTS

1/2/1993 8:00	0.0000000
1/3/1993 8:00	0.0000000
1/4/1993 7:00	0.0000000

WASTE-TREATMENT-WC-3-SOLVENT-MATS-ADDED

1/2/1993 8:00	24.0000000
1/3/1993 8:00	24.0000000
1/4/1993 7:00	24.0000000

WASTE-TREATMENT-WC-3-SOLVENT-QTY-PROCESSED

1/2/1993 8:00	24.0000000
1/3/1993 8:00	24.0000000
1/4/1993 7:00	24.0000000

WASTE-TREATMENT-WC-3-SOLVENT-FEED-STORE

1/2/1993 8:00	0.0000000
1/3/1993 8:00	0.0000000
1/4/1993 7:00	0.0000000

WASTE-TREATMENT-WC-3-STEEL-RECEIPTS

1/2/1993 8:00	4.0000000
1/3/1993 8:00	5.0000000
1/4/1993 7:00	5.0000000

WASTE-TREATMENT-WC-3-STEEL-MATS-ADDED

1/2/1993 8:00	0.0000000
1/3/1993 8:00	0.0000000
1/4/1993 7:00	0.0000000

WASTE-TREATMENT-WC-3-STEEL-QTY-PROCESSED

1/2/1993 8:00	2.0000000
1/3/1993 8:00	2.0000000
1/4/1993 7:00	2.0000000

WASTE-TREATMENT-WC-3-STEEL-FEED-STORE

1/2/1993 8:00	2.0000000
1/3/1993 8:00	3.0000000
1/4/1993 7:00	3.0000000