Transparency Masters for

Going Beyond
An Introduction to
Simulation Using GPSS/H

Unit E:
Arithmetic Expressions,
Synonyms,
S-Functions, E-Functions
Unit E
Remote Arithmetic Expressions, Synonyms, S-Functions, and E-Functions

- Remote Arithmetic Expressions:
  FVARIABLE and VARIABLE Control Statements

- **Model E1**
  The Use of Remote
  Arithmetic Expressions in Model D3

- Synonyms:
  Symbolic Identifiers for Integer Constants,
  and the SYN Compiler Directive

- **Model E2**
  An Application of Synonyms
  in a Manufacturing Model

- S-Functions:
  Functions Whose y-values are Identifiers
  for Entity-Class Members
Model E3
An Application of S-Functions in a Variation of the Model E2 Manufacturing Model

Model E4
A Further Application of S-Functions in a Harbor Model

E-Functions:
Functions Whose y-values are Arithmetic Expressions

Model E5
An Application of E-Functions in a Variation of the Model E4 Harbor Model
Remote Arithmetic Expressions

- As illustrated in preceding models, arithmetic expressions can be used in many contexts, e.g. as Block Operands, as Function arguments, and as Table arguments.

- But when arithmetic expressions are localized in such positions, they cannot be referred to from elsewhere in the model.

- If a given arithmetic expression is to be used in two or more positions in a model, the expression can be provided in the form of a remote arithmetic expression, and in that form can be referred to from any number of different places in the model.

- Remote arithmetic expressions are built into a model by use of the FVARIABLE Control Statement.

  (or in some cases by use of a variation known as a VARIABLE Control Statement)
The FVARIABLE Control Statement

- An FVARIABLE Control Statement has the form:

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>FVARIABLE</td>
<td>arithmetic expression</td>
</tr>
</tbody>
</table>

where:
identifier is the modeler's choice of an identifier for the remote arithmetic expression,

and arithmetic expression is the remote arithmetic expression itself

- An arithmetic expression is composed of SNAs and/or mathematical functions and/or built-in random-variate functions and/or trigonometric functions and/or conversion operators connected by arithmetic operators

- GPSS/H provides these arithmetic operators:

<table>
<thead>
<tr>
<th>Character</th>
<th>Operation</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>multiplication</td>
<td>tied for high</td>
</tr>
<tr>
<td>/</td>
<td>conventional division</td>
<td>tied for high</td>
</tr>
<tr>
<td>@</td>
<td>modulus division</td>
<td>tied for high</td>
</tr>
<tr>
<td>+</td>
<td>addition</td>
<td>tied for low</td>
</tr>
<tr>
<td>-</td>
<td>subtraction</td>
<td>tied for low</td>
</tr>
</tbody>
</table>

- Precedence-level ties are resolved in the usual left-to-right order
<table>
<thead>
<tr>
<th>Name</th>
<th>Value Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS(xpr)</td>
<td>absolute value of xpr</td>
</tr>
<tr>
<td></td>
<td>(xpr: an arithmetic expression)</td>
</tr>
<tr>
<td>ACOS(xpr)</td>
<td>arc cosine of xpr</td>
</tr>
<tr>
<td>ASIN(xpr)</td>
<td>arc sine of xpr</td>
</tr>
<tr>
<td>ATAN(xpr)</td>
<td>arc tangent of xpr</td>
</tr>
<tr>
<td>COS(xpr)</td>
<td>cosine of xpr</td>
</tr>
<tr>
<td>EXP(xpr)</td>
<td>e to the xpr-th power</td>
</tr>
<tr>
<td>FIX(xpr)</td>
<td>the value of xpr converted by truncation to integer</td>
</tr>
<tr>
<td>FLT(xpr)</td>
<td>the value of xpr converted to a floating-point value</td>
</tr>
<tr>
<td>LOG(xpr)</td>
<td>the natural logarithm of xpr</td>
</tr>
<tr>
<td>SIN(xpr)</td>
<td>sine of xpr</td>
</tr>
<tr>
<td>SQRT(xpr)</td>
<td>the square root of xpr</td>
</tr>
<tr>
<td>TAN(xpr)</td>
<td>tangent of xpr</td>
</tr>
</tbody>
</table>
More About arithmetic expressions in FVARIABLE Control Statements

- Use of parentheses is permitted in forming arithmetic expressions

- Arithmetic operations are carried out using floating-point arithmetic

  **Exception**
  
  If the operands on both sides of an arithmetic operator are integer in type, then the operation is carried out using integer arithmetic

  **Example**
  
  $7/4 = 1$, 
  
  whereas $7.0/4.0 = 1.75$, 
  
  $7.0/4 = 1.75$, 
  
  and $7/4.0 = 1.75$, 

- A remote arithmetic expression is invoked by using the syntax:

  $V(identifier)$

  where *identifier* is the identifier used in the corresponding FVARIABLE Control Statement

- The value of an arithmetic expression is returned to the point of invocation in floating-point form (and converted there by truncation to integer if the usage context calls for an integer-type value)
The Basis for Model E1

Model E1 matches the Model D3 Harbor System except that two "immediate" arithmetic expressions in Model D3 are provided as "remote" arithmetic expressions in Model E1.

Two identical arithmetic expressions in Model D3

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATUGWAIT</td>
<td>( (PL(PART1)+AC1-PL(NOTETIME)) \times 60, 0, 5, 12 )</td>
</tr>
<tr>
<td>BTUGWAIT</td>
<td>( (PL(PART1)+AC1-PL(NOTETIME)) \times 60, 0, 5, 12 )</td>
</tr>
</tbody>
</table>

Another two identical arithmetic expressions in Model D3

<table>
<thead>
<tr>
<th>ASSIGN</th>
<th>PART1, AC1-PL(NOTETIME), PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN</td>
<td>PART1, AC1-PL(NOTETIME), PL</td>
</tr>
</tbody>
</table>

(These identical ASSIGN blocks are used in the Type A and the Type B model segments)
Model E1 repeats Model D3

Except that "Immediate" Arithmetic Expressions in Model D3
Are provided in Model E1 in the Alternative Form of
"Remote" Arithmetic Expressions

Base Time Unit: 1 Hour

Compiler Directives (EQU)

NOTETIME EQU 1,PL NOTETIME is Floating-Point Parameter 1
PART1 EQU 2,PL PART1 is Floating-Point Parameter 2

Control Statements (FVARIABLE; STORAGE; TABLE)

...computation of a ship's first wait-time component...
TINWAIT FVARIABLE AC1-PL(NOTETIME)

...computation of a ship's total tug-wait time...
TUGWTIME FVARIABLE (PL(PART1)+AC1-PL(NOTETIME))*60

ABERTHS STORAGE 3 Type A berths
BBERTHS STORAGE 2 Type B berths
TUGBOATS STORAGE 3 tugboats

...total tug wait-time per Type A ship, minutes...
ATUGWAIT TABLE V(TUGWTIME),0,5,12

...total tug wait-time per Type B ship, minutes...
BTUGWAIT TABLE V(TUGWTIME),0,5,12
Model Segment 1 (Type A Ships)

GENERATE 3.2, 1.5, , , 2PL Type A ships arrive
QUEUE AINPORT start AINPORT Queue membership
QUEUE ABERTH start ABERTH Queue membership
ENTER ABERTHS request/capture a Type A berth
DEPART ABERTH end ABERTH Queue membership

...record time at which berthing tugboats are requested...
ASSIGN NOTETIME, AC1, PL

ENTER TUGBOATS, 3 request/capture 3 tugboats

...compute and save the first wait-time component...
ASSIGN PART1, V(TINWAIT), PL

ADVANCE 0.5 berthing time
LEAVE TUGBOATS, 3 give up the 3 tugboats
ADVANCE 8.4, 1.5 un/loading time

...record time at which deberthing tugboats are requested...
ASSIGN NOTETIME, AC1, PL

ENTER TUGBOATS, 2 request/capture 2 tugboats

...tabulate the sum of the two wait-time components...
TABULATE ATUGWAIT

ADVANCE 0.25 deberthing time
LEAVE TUGBOATS, 2 give up the 2 tugboats
LEAVE ABERTHS give up the Type A berth
DEPART AINPORT end AINPORT Queue membership
TERMINATE 1 steam off, reducing the TC by 1
Model Segment 2 (Type B Ships)

GENERATE 1.5,0.75,,2PL Type B ships arrive
QUEUE BINPORT start BINPORT Queue membership
QUEUE BBERTH start BBERTH Queue membership
ENTER BBERTHS request/capture a Type B berth
DEPART BBERTHS end BBERTH Queue membership

...record time at which berthing tugboats are requested...
ASSIGN NOTETIME,AC1,PL

ENTER TUGBOATS,2 request/capture 2 tugboats

...compute and save the first wait-time component...
ASSIGN PART1,V(TINWAIT),PL

ADVANCE 0.5 berthing time
LEAVE TUGBOATS,2 give up the 2 tugboats
ADVANCE 2.1,0.6 un/loading time

...record time at which deberthing tugboats are requested...
ASSIGN NOTETIME,AC1,PL

ENTER TUGBOATS request/capture 1 tugboat

...tabulate the sum of the two wait-time components...
TABULATE BTUGWAIT

ADVANCE 0.25 deberthing time
LEAVE TUGBOATS give up the 1 tugboat
LEAVE BBERTHS give up the Type B berth
DEPART BINPORT end BINPORT Queue membership
TERMINATE 1 steam off, reducing the TC by 1

Run-Control Statements

START 100 set the TC = 100;
start Xact movement
END end of Model-File execution
## Dictionary

**ENTITY DICTIONARY (IN ASCENDING ORDER BY ENTITY NUMBER; "*" = VALUE CONFLICT.)**

| QUEUES: | 1=AINPORT | 2=ABERTH | 3=BINPORT | 4=BBERTH |
| STORAGES: | 1=ABERTHS | 2=BBERTHS | 3=TUGBOATS |
| TABLES: | 1=ATUGWAIT | 2=BTUGWAIT |
| (F) VARIABLES: | 1=TUGWTIME | 2=TINWAIT |
| FLOAT PARMs: | 1=NOTETIME | 2=PART1 |

## Cross-Reference Listing

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>VALUE</th>
<th>EQU DEFNS</th>
<th>CONTEXT</th>
<th>REFERENCES BY STATEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABERTH</td>
<td>2</td>
<td>QUEUE</td>
<td></td>
<td>44</td>
</tr>
<tr>
<td>AINPORT</td>
<td>1</td>
<td>QUEUE</td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>BBERTH</td>
<td>4</td>
<td>QUEUE</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>BINPORT</td>
<td>3</td>
<td>QUEUE</td>
<td></td>
<td>79</td>
</tr>
<tr>
<td>ABERTHS</td>
<td>1</td>
<td>STORAGE</td>
<td>22</td>
<td>45</td>
</tr>
<tr>
<td>BBERTHS</td>
<td>2</td>
<td>STORAGE</td>
<td>23</td>
<td>81</td>
</tr>
<tr>
<td>TUGBOATS</td>
<td>3</td>
<td>STORAGE</td>
<td>24</td>
<td>51</td>
</tr>
<tr>
<td>ATUGWAIT</td>
<td>1</td>
<td>TABLE</td>
<td>27</td>
<td>66</td>
</tr>
<tr>
<td>BTUGWAIT</td>
<td>2</td>
<td>TABLE</td>
<td>30</td>
<td>102</td>
</tr>
<tr>
<td>TINWAIT</td>
<td>2</td>
<td>(F) VARIABLE</td>
<td>36</td>
<td>54</td>
</tr>
<tr>
<td>TUGWTIME</td>
<td>1</td>
<td>(F) VARIABLE</td>
<td>33</td>
<td>27</td>
</tr>
<tr>
<td>NOTETIME</td>
<td>1</td>
<td>FLOAT PAR</td>
<td>15</td>
<td>33</td>
</tr>
<tr>
<td>PART1</td>
<td>2</td>
<td>FLOAT PAR</td>
<td>16</td>
<td>33</td>
</tr>
</tbody>
</table>
VARIABLE Control Statements
(a variation of FVARIABLE Control Statements)

- If the Operation word VARIABLE
  (instead of FVARIABLE)
  is used in the Control Statement for a remote
  arithmetic expression, then the expression will
  be evaluated using integer-type arithmetic

- This means floating-point data elements in the
  arithmetic expression will have their values
  converted by truncation to integer before
  arithmetic operations are performed on them

- Remote arithmetic expressions defined by using
  VARIABLE as the Operation are invoked with the
  syntax:

  \[ V(\text{identifier}) \]

  where \text{identifier} is the identifier used in the
  corresponding VARIABLE Control Statement

  (Recall that the \text{V(identifier)} referencing pattern is
  also used to refer to remote arithmetic expressions
  defined by using FVARIABLE as the Operation)
A Useful Application for Integer-Type Arithmetic
(that is, for VARIABLE Control Statements)

- It is sometimes necessary to relate absolute time (AC1) to relative time in a simulation.

- Absolute time (AC1) grows in value as a simulation proceeds, so this sets up the need to convert the value of AC1 into a corresponding relative-time value.

- For example, it might be necessary to know:
  ✓ which minute it is within the current hour
  ✓ which hour it is within the current shift
  ✓ which shift it is within the current week
  ✓ which week it is within the current month
  ✓ which month it is within the current year
  etc.

- Appropriately defined remote arithmetic expressions of the VARIABLE type can be used for this purpose, as the following examples show.
VARIABLE Control Statements
Relating AC1 to Relative Time
(Examples)

- Assumptions:
  ✓ Base Time Unit: 1 minute
  ✓ Time is simulated from 8 a.m. until 4 p.m.
    each weekday (1 shift/day, 5 days/week)
  ✓ AC1 = 0.0 corresponds to 8 a.m. on day 1

  (then AC1 = 60.0 is 9 a.m. for the day 1 shift;
   AC1 = 480.0 is both 4 p.m. for the day 1 shift
   and 8 a.m. for the day 2 shift;
   AC1 = 540.0 is 9 a.m. for the day 2 shift;
   AC1 = 2400.0 is both 4 p.m. for the day 5 shift
   and 8 a.m. for the day 6 shift, etc.)

EXAMPELE1 VARIABLE AC1/480

...no. of shifts completed...

EXAMPELE2 VARIABLE AC1/2400

...no. of 5-day weeks completed...

EXAMPELE3 VARIABLE AC1@60

...no. of minutes completed within the current hour...

EXAMPELE4 VARIABLE (AC1@480)/60

...no. of hours completed within the current shift...

EXAMPELE5 VARIABLE (AC1@2400)/480
Synonyms in GPSS/H:
The SYN Compiler Directive

- Synonyms are *symbolic identifiers* for *integer constants*

- Synonyms are *not identifiers for entity-class members*  
  (they are identifiers that apply outside the context of entity classes)

- Synonyms are built into a model by use of the SYN Compiler Directive

- In statement terms, the SYN Compiler Directive takes this form:

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>SYN</td>
<td>j</td>
</tr>
</tbody>
</table>

  where *identifier* is the modeler's choice of an identifier for the nonnegative integer constant *j*

- Examples of SYN Compiler Directives

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVENCOST</td>
<td>SYN</td>
<td>80 oven cost is $80/day</td>
</tr>
<tr>
<td>DAYWAGE</td>
<td>SYN</td>
<td>75 the daily wage is $75</td>
</tr>
<tr>
<td>JOBTYPE1</td>
<td>SYN</td>
<td>1 JOBTYPE1 has 1 as its value</td>
</tr>
<tr>
<td>JOBTYPE2</td>
<td>SYN</td>
<td>2 JOBTYPE2 has 2 as its value</td>
</tr>
</tbody>
</table>
Synonyms in GPSS/H
(continued)

- Synonyms can be used in place of integer constants anywhere in a GPSS/H model, e.g.,
  ✓ as the x coordinates in Functions
  ✓ as Block Operands
  ✓ as elements in arithmetic expressions

- Synonyms are not variables
  (Their values are constant; unlike the values of Ampervariables, the values of Synonyms cannot be changed by Control-Statement execution, or by Block execution, or in any other way except by editing SYN Compiler Directives in the GPSS/H Model File and then recompiling the model)

- Why use Synonyms?
  ✓ to provide memory-aiding support in model-building
  ✓ to enhance the readability of models
  ✓ to facilitate the process of applying global changes to model parameters in some cases
    (editing a SYNONYM Compiler Directive and recompiling a Model File is easier and less error-prone than changing the value of an integer constant in multiple locations throughout the Model File)
Flow Schematic for Models E2 and E3

Orders Arrive for Products of Type 1, Type 2, and Type 3

Three Identical Drills

Two Identical Lathes

Four Identical Mills

Finished Orders Leave

Exponentially Distributed Order Interarrival Times (Mean: 30 Minutes)

30% Type 1 orders, 50% Type 2 orders, 20% Type 3 orders

Step Times
Type 1: 40 Minutes
Type 2: 115 Minutes
Type 3: 90 Minutes

Step Times
Type 1: 25 Minutes
Type 2: 60 Minutes
Type 3: 50 Minutes

Step Times
Type 1: 60 Minutes
Type 2: 135 Minutes
Type 3: 120 Minutes
Three Types of Product Are Made in a Machining Sequence (Drill-Mill-Lathe) Common to All Three Product Types; Machining Times, However, Are Specific to the Type of Product and the Type of Machine

Synonyms Are Used to Give Symbolic Names to Each Product Type (Synonyms Are Symbolic Names for Integer Constants)

Each Product-Xact Carries (the Numeric Equivalent of) its Product Name in a Fullword Parameter

Functions Provide Machining Times as a Function of Product Type

A Single Stack of Blocks is Used by Each Product Type (As a Result, Fewer Blocks Are Needed Than if Each Product Type Used Its Own Private Stack of Blocks)

Order Fill-Time is Consolidated in One Table for All Three Product Types

Base Time Unit: 1 Minute

Compiler Directives (EQU; SYN)

PROTYPE EQU 1,PF PROTYPE is Fullword Parameter 1

TYPE1 SYN 1 TYPE1 is a synonym for the number 1
TYPE2 SYN 2 TYPE2 " " " 2
TYPE3 SYN 3 TYPE3 " " " 3

Control Statements (FUNCTION; STORAGE; TABLE)

PRODUCTS FUNCTION RN1,D3 product-type distribution .3,TYPE1/.8,TYPE2/1,TYPE3

DRILLTIME FUNCTION PF(PRODTYPE),D3 drilling time as a function of product type TYPE1,40/TYPD2,115/TYPD3,90

LATHTIME FUNCTION PF(PRODTYPE),D3 lathe time as a function of product type TYPE1,25/TYPD2,60/TYPD3,50

MILLTIME FUNCTION PF(PRODTYPE),D3 milling time as a function of product type TYPE1,60/TYPD2,135/TYPD3,120

DRILLS STORAGE 3 number of drills
LATHES STORAGE 2 number of lathes
MILLS STORAGE 4 number of mills

FILLTIME TABLE M1/60,4,4,6 consolidated order fill-time, hours
Model Segment 1 (Movement of Orders Through the Shop)

GENERATE RVEXPO(1,30.0),,,1PF product orders arrive

...randomly determine type of product ordered...
ASSIGN PRODTYPE,FN(PRODTYPES),PF

ENTER DRILLS request/capture a drill
ADVANCE FN(DRILTIME) drilling time
LEAVE DRILLS let this drill go

ENTER LATHES request/capture a lathe
ADVANCE FN(LATHTIME) lathing time
LEAVE LATHES let this lathe go

ENTER MILLS request/capture a mill
ADVANCE FN(MILLTIME) milling time
LEAVE MILLS let this mill go

TABULATE FILLTIME tabulate order fill-time

TERMINATE 1 finished product leaves the shop

Run-Control Statements

START 500 start Xact movement
END end of Model-File execution
ENTITY DICTIONARY (IN ASCENDING ORDER BY ENTITY NUMBER; "*" = VALUE CONFLICT.)

STORAGES: 1=DRILLS  2=LATHES  3=MILLS

TABLES: 1=FILLTIME

FUNCTIONS: 1=PRODTYPES  2=DRILLTIME  3=LATHTIME  4=MILLTIME

FULLWORD PARMs: 1=PRODTYPE

RANDOM NUMBERS: 1

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>VALUE</th>
<th>EQU</th>
<th>DEFNS</th>
<th>CONTEXT</th>
<th>REFERENCES BY STATEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE1</td>
<td>1</td>
<td>32</td>
<td></td>
<td>ABSOLUTE</td>
<td>41 44 47 50</td>
</tr>
<tr>
<td>TYPE2</td>
<td>2</td>
<td>33</td>
<td></td>
<td>ABSOLUTE</td>
<td>41 44 47 50</td>
</tr>
<tr>
<td>TYPE3</td>
<td>3</td>
<td>34</td>
<td></td>
<td>ABSOLUTE</td>
<td>41 44 47 50</td>
</tr>
<tr>
<td>DRILLS</td>
<td>1</td>
<td>52</td>
<td>STORAGE</td>
<td></td>
<td>67 69</td>
</tr>
<tr>
<td>LATHES</td>
<td>2</td>
<td>53</td>
<td>STORAGE</td>
<td></td>
<td>71 73</td>
</tr>
<tr>
<td>MILLS</td>
<td>3</td>
<td>54</td>
<td>STORAGE</td>
<td></td>
<td>75 77</td>
</tr>
<tr>
<td>FILLTIME</td>
<td>1</td>
<td>56</td>
<td>TABLE</td>
<td></td>
<td>79</td>
</tr>
<tr>
<td>DRILLTIME</td>
<td>2</td>
<td>43</td>
<td>FUNCTION</td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>LATHTIME</td>
<td>3</td>
<td>46</td>
<td>FUNCTION</td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>MILLTIME</td>
<td>4</td>
<td>49</td>
<td>FUNCTION</td>
<td></td>
<td>76</td>
</tr>
<tr>
<td>PRODTYPES</td>
<td>1</td>
<td>40</td>
<td>FUNCTION</td>
<td></td>
<td>65</td>
</tr>
<tr>
<td>PRODTYPE</td>
<td>1</td>
<td>30</td>
<td>FULLWORD PAR</td>
<td></td>
<td>43 46 49 65</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>RANDOM #</td>
<td></td>
<td>40 62</td>
</tr>
</tbody>
</table>
RELATIVE CLOCK:  15674.2404  ABSOLUTE CLOCK:  15674.2404

<table>
<thead>
<tr>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
<th>BLOCK CURRENT</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>521</td>
<td>11</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>12</td>
<td>500</td>
</tr>
<tr>
<td>3</td>
<td>507</td>
<td>13</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
<td>507</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
<td>504</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td></td>
<td>504</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td></td>
<td>504</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td></td>
<td>502</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td></td>
<td>502</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td></td>
<td>502</td>
</tr>
</tbody>
</table>

--AVG-UTIL-DURING--

<table>
<thead>
<tr>
<th>STORAGE</th>
<th>TOTAL</th>
<th>AVAIL</th>
<th>UNAVL</th>
<th>ENTRIES</th>
<th>AVERAGE</th>
<th>CURRENT</th>
<th>PERCENT</th>
<th>CAPACITY</th>
<th>AVERAGE</th>
<th>CURRENT</th>
<th>MAXIMUM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TIME</td>
<td>TIME</td>
<td>TIME</td>
<td></td>
<td>TIME/UNIT</td>
<td>STATUS</td>
<td>AVAIL</td>
<td></td>
<td>CONTENTS</td>
<td>CONTENTS</td>
<td>CONTENTS</td>
</tr>
<tr>
<td>DRILLS</td>
<td>.961</td>
<td>507</td>
<td>89.141</td>
<td>100.0</td>
<td>3</td>
<td>2.883</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LATHE</td>
<td>.775</td>
<td>504</td>
<td>48.235</td>
<td>100.0</td>
<td>2</td>
<td>1.551</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MILLS</td>
<td>.887</td>
<td>502</td>
<td>110.788</td>
<td>100.0</td>
<td>4</td>
<td>3.548</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE FILLTIME

<table>
<thead>
<tr>
<th>ENTRIES IN TABLE</th>
<th>MEAN ARGUMENT</th>
<th>STANDARD DEVIATION</th>
<th>SUM OF ARGUMENTS</th>
<th>NON-WEIGHTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>500.0000</td>
<td>9.4625</td>
<td>4.2122</td>
<td>4731.2374</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPPER LIMIT</th>
<th>OBSERVED FREQ</th>
<th>PERCENT OF TOTAL</th>
<th>CUMULATIVE PERCENTAGE</th>
<th>CUMULATIVE REMAINDER OF MEAN</th>
<th>MULTIPLE OF MEAN</th>
<th>DEVIATION FROM MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0000</td>
<td>29.0000</td>
<td>5.80</td>
<td>5.80</td>
<td>94.20</td>
<td>0.4227</td>
<td>-1.2968</td>
</tr>
<tr>
<td>8.0000</td>
<td>197.0000</td>
<td>39.40</td>
<td>45.20</td>
<td>54.80</td>
<td>0.8454</td>
<td>-0.3472</td>
</tr>
<tr>
<td>12.0000</td>
<td>134.0000</td>
<td>26.80</td>
<td>72.00</td>
<td>28.00</td>
<td>1.2682</td>
<td>0.6024</td>
</tr>
<tr>
<td>16.0000</td>
<td>94.0000</td>
<td>18.80</td>
<td>90.80</td>
<td>9.20</td>
<td>1.6909</td>
<td>1.5521</td>
</tr>
<tr>
<td>20.0000</td>
<td>46.0000</td>
<td>9.20</td>
<td>100.00</td>
<td>0.00</td>
<td>2.1136</td>
<td>2.5017</td>
</tr>
</tbody>
</table>

...
Indirectly Supplying Identifiers for Queues, Facilities, Storages, Tables, etc.: S-Functions

- In models of this form:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE</td>
<td>35,15</td>
</tr>
<tr>
<td>QUEUE</td>
<td>DRILLQUE</td>
</tr>
<tr>
<td>SEIZE</td>
<td>DRILL</td>
</tr>
<tr>
<td>DEPART</td>
<td>DRILLQUE</td>
</tr>
<tr>
<td>ADVANCE</td>
<td>30,10</td>
</tr>
<tr>
<td>RELEASE</td>
<td>DRILL</td>
</tr>
<tr>
<td>TABULATE</td>
<td>SYSTIME</td>
</tr>
<tr>
<td>TERMINATE</td>
<td>1</td>
</tr>
</tbody>
</table>

Identifiers for members of entity classes such as Queues, Facilities, Tables, etc. are supplied directly in Block-Operand positions.

- It is also possible to supply an identifier for a member of an entity class indirectly by specifying in the relevant Block-Operand position a Function that returns the identifier as its value.

- The type of Function used for this purpose is an S-Function.
The Format of S-Functions

- Like D-Functions and C-Functions, a modeler defines S-Functions with a FUNCTION Control Statement, followed by one or more Function Follower Statements.

- The general format of these two types of statements in the setting of S-Functions is shown below:

\[
\text{label} \quad \text{FUNCTION} \quad \text{argument, type}_\text{count}, \text{entity}_\text{class} \quad \text{FUNCTION} \quad \text{Control Statement}
\]

\[
\text{FUNCTION} \quad \text{Control Statement} \quad \text{Function Follower Statement}
\]

\[
X_1, y_1 / X_2, y_2 / X_3, y_3 / \ldots / X_\text{count}, y_\text{count}
\]

where: \text{label}, \text{argument} and \text{count} have their usual significance,

\text{type} is the letter \text{S},

\text{entity}_\text{class}

is a code indicating the entity class for which the y-elements of the pairs of points are member identifiers,

and the y elements in the x,y-pairs are identifiers for members of the indicated entity class.
An Example of an S-Function

- In the context of Model E3, here is an example of a possible S-Function:

<table>
<thead>
<tr>
<th>TABINDEX</th>
<th>FUNCTION</th>
<th>PF(PRODTYPE),S3,T</th>
<th>TYPE1,TYME1TYM/TYPE2,TYME2TYM/TYPE3,TYME3TYM</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE1</td>
<td>SYN</td>
<td>1</td>
<td>TYPE1 is a synonym for 1</td>
</tr>
<tr>
<td>TYPE2</td>
<td>SYN</td>
<td>2</td>
<td>TYPE2 is a synonym for 2</td>
</tr>
<tr>
<td>TYPE3</td>
<td>SYN</td>
<td>3</td>
<td>TYPE3 is a synonym for 3</td>
</tr>
<tr>
<td>TYPE1TYM</td>
<td>TABLE</td>
<td>M1/60,4,4,6</td>
<td>Type 1 order fill-time</td>
</tr>
<tr>
<td>TYPE2TYM</td>
<td>TABLE</td>
<td>M1/60,4,4,6</td>
<td>Type 2 order fill-time</td>
</tr>
<tr>
<td>TYPE3TYM</td>
<td>TABLE</td>
<td>M1/60,4,4,6</td>
<td>Type 3 order fill-time</td>
</tr>
</tbody>
</table>

- In this example, the C-Operand T in the FUNCTION Control Statement indicates that the Function's y-values (TYPE1TYM, TYPE2TYM, and TYPE3TYM) are identifiers for Tables.

- The TABINDEX S-Function could be referred in the model by having order-Xacts execute the following TABULATE Block just before leaving the model:

  TABULATE FN(TABINDEX)

  The effect would be to tabulate order fill-time by order type (one Table per type of order)
## Entity-Type Codes for the C-Operand in S-Functions

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facility</td>
<td>F</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>Byte</td>
<td>PB</td>
</tr>
<tr>
<td>Halfword</td>
<td>PH</td>
</tr>
<tr>
<td>Floating Point</td>
<td>PL</td>
</tr>
<tr>
<td>Fullword</td>
<td>PF</td>
</tr>
<tr>
<td>Queue</td>
<td>Q</td>
</tr>
<tr>
<td>Random Number Stream</td>
<td>RN</td>
</tr>
<tr>
<td>Storage</td>
<td>S</td>
</tr>
<tr>
<td>Table</td>
<td>T</td>
</tr>
<tr>
<td>Variable</td>
<td>V</td>
</tr>
<tr>
<td>Function</td>
<td>Z</td>
</tr>
</tbody>
</table>

### Codes for Other Entity Types Not Yet Discussed

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean Variable</td>
<td>B</td>
</tr>
<tr>
<td>Group</td>
<td>G</td>
</tr>
<tr>
<td>Logic Switch</td>
<td>L</td>
</tr>
<tr>
<td>Savevalues (linear list)</td>
<td></td>
</tr>
<tr>
<td>Byte</td>
<td>XB</td>
</tr>
<tr>
<td>Halfword</td>
<td>XH</td>
</tr>
<tr>
<td>Floating Point</td>
<td>XL</td>
</tr>
<tr>
<td>Fullword</td>
<td>XF</td>
</tr>
<tr>
<td>Savevalues (matrix)</td>
<td></td>
</tr>
<tr>
<td>Byte</td>
<td>MB</td>
</tr>
<tr>
<td>Halfword</td>
<td>MH</td>
</tr>
<tr>
<td>Floating Point</td>
<td>ML</td>
</tr>
<tr>
<td>Fullword</td>
<td>MX</td>
</tr>
<tr>
<td>User Chain</td>
<td>C</td>
</tr>
</tbody>
</table>
Flow Schematic for Models E2 and E3

Orders Arrive for Products of Type 1, Type 2, and Type 3

Exponentially Distributed Order Interarrival Times (Mean: 30 Minutes)
- 30% Type 1 orders,
- 50% Type 2 orders,
- 20% Type 3 orders

Three Identical Drills

Step Times
- Type 1: 40 Minutes
- Type 2: 115 Minutes
- Type 3: 90 Minutes

Two Identical Lathes

Step Times
- Type 1: 25 Minutes
- Type 2: 60 Minutes
- Type 3: 50 Minutes

Four Identical Mills

Step Times
- Type 1: 60 Minutes
- Type 2: 135 Minutes
- Type 3: 120 Minutes

Finished Orders Leave
Model E3 is Identical to Model E2
Except that S-Functions Are Used
to Tabulate Order Fill-Times by Product Type

(As a Result, There are 3 Tables in the Model,
one Table per Product Type)

(The y-values of S-Functions Are
Symbolic Identifiers for Members of an Entity Class)

Base Time Unit: 1 Minute

Compiler Directives (EQU; SYN)

PRODTYPE EQU 1,PF PRODTYPE is Fullword Parameter 1

TYPE1 SYN 1 TYPE1 is a synonym for the number 1
TYPE2 SYN 2 TYPE2 " " 2
TYPE3 SYN 3 TYPE3 " " 3

Control Statements (FUNCTION; STORAGE; TABLE)

PRODTPS FUNCTION RN1,D3 product-type distribution
.3,TYPE1/.8,TYPE2/1,TYPE3

DRILTIME FUNCTION PF(PRODTYPE),D3 drilling time as a function of product type
TYPE1,40/TYPE2,115/TYPE3,90

LATHTIME FUNCTION PF(PRODTYPE),D3 lathe time as a function of product type
TYPE1,25/TYPE2,60/TYPE3,50

MILLTIME FUNCTION PF(PRODTYPE),D3 milling time as a function of product type
TYPE1,60/TYPE2,135/TYPE3,120

TABINDEX FUNCTION PF(PRODTYPE),S3,T Table identifiers as a function of order type
TYPE1,TYPE1TYM/TYPE2,TYPE2TYM/TYPE3,TYPE3TYM

DRILLS STORAGE 3 number of drills
LATHES STORAGE 2 number of lathes
MILLS STORAGE 4 number of mills

TYPE1TYM TABLE M1/60,4,4,6 Type 1 order fill-time, hours
TYPE2TYM TABLE M1/60,4,4,6 Type 2 order fill-time, hours
TYPE3TYM TABLE M1/60,4,4,6 Type 3 order fill-time, hours
Model Segment 1 (Movement of Orders Through the Shop)

GENERATE RVEXPO(1,30.0),,,1PF  product orders arrive

...randomly determine type of product ordered...
ASSIGN PRODTYPE, FN(PRODTYPES),PF

ENTER DRILLS request/capture a drill
ADVANCE FN(DRILTIME) drilling time
LEAVE DRILLS let this drill go

ENTER LATHES request/capture a lathe
ADVANCE FN(LATHTIME) lathing time
LEAVE LATHES let this lathe go

ENTER MILLS request/capture a mill
ADVANCE FN(MILLTIME) milling time
LEAVE MILLS let this mill go

TABULATE FN(TABINDEX) tabulate order fill-time
by type of order
TERMINATE 1 finished order leaves the shop

Run-Control Statements

START 500 start Xact movement
END end of Model-File execution
## Cross-Reference Listing
for Model E3

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>VALUE</th>
<th>EQU</th>
<th>DEFINES</th>
<th>CONTEXT</th>
<th>REFERENCES BY STATEMENT NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE1</td>
<td>1</td>
<td>21</td>
<td></td>
<td>ABSOLUTE</td>
<td>30 33 36 39 42</td>
</tr>
<tr>
<td>TYPE2</td>
<td>2</td>
<td>22</td>
<td></td>
<td>ABSOLUTE</td>
<td>30 33 36 39 42</td>
</tr>
<tr>
<td>TYPE3</td>
<td>3</td>
<td>23</td>
<td></td>
<td>ABSOLUTE</td>
<td>30 33 36 39 42</td>
</tr>
<tr>
<td>DRILLS</td>
<td>1</td>
<td></td>
<td>45</td>
<td>STORAGE</td>
<td>62 64</td>
</tr>
<tr>
<td>LATHES</td>
<td>2</td>
<td></td>
<td>46</td>
<td>STORAGE</td>
<td>66 68</td>
</tr>
<tr>
<td>MILLS</td>
<td>3</td>
<td></td>
<td>47</td>
<td>STORAGE</td>
<td>70 72</td>
</tr>
<tr>
<td>TYPE1TYM</td>
<td>1</td>
<td></td>
<td>49</td>
<td>TABLE</td>
<td>42</td>
</tr>
<tr>
<td>TYPE2TYM</td>
<td>2</td>
<td></td>
<td>50</td>
<td>TABLE</td>
<td>42</td>
</tr>
<tr>
<td>TYPE3TYM</td>
<td>3</td>
<td></td>
<td>51</td>
<td>TABLE</td>
<td>42</td>
</tr>
<tr>
<td>DRILLTIME</td>
<td>2</td>
<td></td>
<td>32</td>
<td>FUNCTION</td>
<td>63</td>
</tr>
<tr>
<td>LATHTIME</td>
<td>3</td>
<td></td>
<td>35</td>
<td>FUNCTION</td>
<td>67</td>
</tr>
<tr>
<td>MILLTIME</td>
<td>4</td>
<td></td>
<td>38</td>
<td>FUNCTION</td>
<td>71</td>
</tr>
<tr>
<td>PRODTYPE</td>
<td>1</td>
<td></td>
<td>29</td>
<td>FUNCTION</td>
<td>60</td>
</tr>
<tr>
<td>TABINDEX</td>
<td>5</td>
<td></td>
<td>41</td>
<td>FUNCTION</td>
<td>74</td>
</tr>
<tr>
<td>PRODTYPE</td>
<td>1</td>
<td></td>
<td>19</td>
<td>FULLWORD</td>
<td>32 35 38 41 60</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>RANDOM #</td>
<td>29 57</td>
</tr>
</tbody>
</table>
### TABLE TYPE1TYM

<table>
<thead>
<tr>
<th>ENTRIES IN TABLE</th>
<th>MEAN</th>
<th>ARGUMENT</th>
<th>STANDARD DEVIATION</th>
<th>SUM OF ARGUMENTS</th>
<th>NON-WEIGHTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>140.0000</td>
<td>7.6814</td>
<td>4.0866</td>
<td>1075.3998</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPPER LIMIT</th>
<th>OBSERVED FREQUENCY</th>
<th>OBSERVED PERCENT OF TOTAL</th>
<th>OBSERVED PERCENTAGE</th>
<th>CUMULATIVE PERCENTAGE</th>
<th>CUMULATIVE REMAINDER</th>
<th>MULTIPLE DEVIATION OF MEAN</th>
<th>DEVIATION FROM MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0000</td>
<td>29.0000</td>
<td>20.71</td>
<td>20.71</td>
<td>79.29</td>
<td>0.5207</td>
<td>-0.9009</td>
<td></td>
</tr>
<tr>
<td>8.0000</td>
<td>58.0000</td>
<td>41.43</td>
<td>62.14</td>
<td>37.86</td>
<td>1.0415</td>
<td>0.0780</td>
<td></td>
</tr>
<tr>
<td>12.0000</td>
<td>20.0000</td>
<td>14.29</td>
<td>76.43</td>
<td>23.57</td>
<td>1.5622</td>
<td>1.0568</td>
<td></td>
</tr>
<tr>
<td>16.0000</td>
<td>30.0000</td>
<td>21.43</td>
<td>97.86</td>
<td>2.14</td>
<td>2.0829</td>
<td>2.0356</td>
<td></td>
</tr>
<tr>
<td>20.0000</td>
<td>3.0000</td>
<td>2.14</td>
<td>100.00</td>
<td>0.00</td>
<td>2.6037</td>
<td>3.0144</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE TYPE2TYM

<table>
<thead>
<tr>
<th>ENTRIES IN TABLE</th>
<th>MEAN</th>
<th>ARGUMENT</th>
<th>STANDARD DEVIATION</th>
<th>SUM OF ARGUMENTS</th>
<th>NON-WEIGHTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>262.0000</td>
<td>10.5076</td>
<td>3.9840</td>
<td>2752.9781</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPPER LIMIT</th>
<th>OBSERVED FREQUENCY</th>
<th>OBSERVED PERCENT OF TOTAL</th>
<th>OBSERVED PERCENTAGE</th>
<th>CUMULATIVE PERCENTAGE</th>
<th>CUMULATIVE REMAINDER</th>
<th>MULTIPLE DEVIATION OF MEAN</th>
<th>DEVIATION FROM MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.0000</td>
<td>86.0000</td>
<td>32.82</td>
<td>32.82</td>
<td>67.18</td>
<td>0.7614</td>
<td>-0.6294</td>
<td></td>
</tr>
<tr>
<td>12.0000</td>
<td>94.0000</td>
<td>35.88</td>
<td>68.70</td>
<td>31.30</td>
<td>1.1420</td>
<td>0.3746</td>
<td></td>
</tr>
<tr>
<td>16.0000</td>
<td>48.0000</td>
<td>18.32</td>
<td>87.02</td>
<td>12.98</td>
<td>1.5227</td>
<td>1.3786</td>
<td></td>
</tr>
<tr>
<td>20.0000</td>
<td>34.0000</td>
<td>12.98</td>
<td>100.00</td>
<td>0.00</td>
<td>1.9034</td>
<td>2.3827</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE TYPE3TYM

<table>
<thead>
<tr>
<th>ENTRIES IN TABLE</th>
<th>MEAN</th>
<th>ARGUMENT</th>
<th>STANDARD DEVIATION</th>
<th>SUM OF ARGUMENTS</th>
<th>NON-WEIGHTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>98.0000</td>
<td>9.2129</td>
<td>4.1283</td>
<td>902.8595</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPPER LIMIT</th>
<th>OBSERVED FREQUENCY</th>
<th>OBSERVED PERCENT OF TOTAL</th>
<th>OBSERVED PERCENTAGE</th>
<th>CUMULATIVE PERCENTAGE</th>
<th>CUMULATIVE REMAINDER</th>
<th>MULTIPLE DEVIATION OF MEAN</th>
<th>DEVIATION FROM MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.0000</td>
<td>53.0000</td>
<td>54.08</td>
<td>54.08</td>
<td>45.92</td>
<td>0.8684</td>
<td>-0.2938</td>
<td></td>
</tr>
<tr>
<td>12.0000</td>
<td>20.0000</td>
<td>20.41</td>
<td>74.49</td>
<td>25.51</td>
<td>1.3025</td>
<td>0.6751</td>
<td></td>
</tr>
<tr>
<td>16.0000</td>
<td>16.0000</td>
<td>16.33</td>
<td>90.82</td>
<td>9.18</td>
<td>1.7367</td>
<td>1.6440</td>
<td></td>
</tr>
<tr>
<td>20.0000</td>
<td>9.0000</td>
<td>9.18</td>
<td>100.00</td>
<td>0.00</td>
<td>2.1709</td>
<td>2.6130</td>
<td></td>
</tr>
</tbody>
</table>

...
Schematic for the Model E4 Harbor System

Type A Ship

Type B Ship

Type C Ship

Type C Berth

Tugboat

Tugboat

Tugboat

Tugboat

Tugboat

Type C Berth

Type C Berth

Type C Berth

Type A Berth

Type B Berth

Type A Berth

Type B Berth

Type A Berth
*************** Model E4 ***********************
SIMULATE Ships Un/loading Cargo at a Harbor
(Three Types of Ships, Three Types of Berths)

Model E4 Applies the Techniques and Procedures of Models E2 and E3
(All Three Types of Ships Move Through a Single Stack of Blocks,
Achieving Economy in the Number of Blocks Needed)

Model E4 Goes Beyond Models E2 and E3, however, in that
S-Functions Are Used to Refer Indirectly to Resource Types (Berths)
in Model E4 (there was no occasion to do this in Models E2 or E3)

Base Time Unit: 1 Hour

Compiler Directives (EQU; SYN; UNLIST)

SType EQU 1,PF SType is Fullword Parameter 1
ASHIP SYN 1 ASHIP is a synonym for 1
BSHIP SYN 2 BSHIP is a synonym for 2
CSHIP SYN 3 CSHIP is a synonym for 3

UNLIST CSECHO suppress Control-Statement echoes

Control Statements (FUNCTION; QTABLE; STORAGE)

BRTHTYPE FUNCTION PF(SType),S3,S berth-type identifier as
ASHIP,ABERTHS/BSHIP,BBERTHS/CSHIP,CBERTHS a function of ship type

IMPORTQ FUNCTION PF(SType),S3,Q in-port Queue identifier
ASHIP,AIMPORTQ/BSHIP,BIMPORTQ/CSHIP,CIMPORTQ as function of ship type

STYPES FUNCTION RN1,D3 .45,ASHIP/.65,BSHIP/1,CSHIP
distribution of
types of ships

TUGSIN FUNCTION PF(SType),D3 ASHIP,3/BSHIP,5/CSHIP,4
number of berthing tugs needed
as a function of ship type

TUGSOUT FUNCTION PF(SType),D3 ASHIP,1/BSHIP,3/CSHIP,2
number of deberthing tugs needed
as a function of ship type

ULMEANS FUNCTION PF(SType),D3 ASHIP,20/BSHIP,31/CSHIP,26
mean unloading time
as a function of ship type

ULSDEVS FUNCTION PF(SType),D3 ASHIP,4/BSHIP,6/CSHIP,5
unloading time std deviation
as a function of ship type

AIMPORT QTABLE AIMPORTQ,12,12,8 distributions of in-port
BINPORT QTABLE BIMPORTQ,12,12,10 residence times for the
CIMPORT QTABLE CIMPORTQ,12,12,8 three types of ships

ABERTHS STORAGE 3 Type A berths
BBERTHS STORAGE 2 Type B berths
CBERTHS STORAGE 4 Type C berths

TUGBOATS STORAGE 7 tugboats
Model Segment 1 (Movement of Ships Through the Harbor)

...ships arrive...
GENERATE RVEXPO(1,3.9),,,1PF

...let PF(STYPE) = type of ship
ASSIGN STYPE,FN(STYPES),PF

...start in-port Queue membership...
QUEUE FN(INPORTQ)

...request/capture berth-type needed...
ENTER FN(BRTHYPE)

...request/capture tugboats needed for berthing...
ENTER TUGBOATS,FN(TUGSIN)

...berthing time...
ADVANCE 1.0

...give up the tugboats used for berthing...
LEAVE TUGBOATS,FN(TUGSIN)

...normally distributed un/loading times...
ADVANCE ABS(RVNORM(1,FN(ULMEANS),FN(ULSDEVs)))

...request/capture tugboats needed for debberthing...
ENTER TUGBOATS,FN(TUGSOUT)

...deberthing time...
ADVANCE 0.5

...give up the tugboats used for debberthing...
LEAVE TUGBOATS,FN(TUGSOUT)

...give up the berth...
LEAVE FN(BRTHYPE)

...end in-port Queue membership...
DEPART FN(INPORTQ)

...steam off...
TERMINATE 1

Run-Control Statements

START 100,NP initialize with 100 ships;
No Printing of standard report

RESET reinitialize the model statistically

START 500 base steady-state results on
500 ships

END end of Model-File execution
### Table AINPORT

<table>
<thead>
<tr>
<th>Entries in Table</th>
<th>Mean Argument</th>
<th>Standard Deviation</th>
<th>Sum of Arguments</th>
<th>Non-Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>207.0000</td>
<td>27.3813</td>
<td>9.9738</td>
<td>5667.9188</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upper Limit</th>
<th>Observed Frequency</th>
<th>Percent of Total</th>
<th>Percentage Cumulative</th>
<th>Cumulative Remainder of Mean</th>
<th>Multiple of Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.0000</td>
<td>4.0000</td>
<td>1.93</td>
<td>1.93</td>
<td>98.07</td>
<td>0.4383</td>
</tr>
<tr>
<td>24.0000</td>
<td>89.0000</td>
<td>43.00</td>
<td>44.93</td>
<td>55.07</td>
<td>0.8765</td>
</tr>
<tr>
<td>36.0000</td>
<td>85.0000</td>
<td>41.06</td>
<td>85.99</td>
<td>14.01</td>
<td>1.3148</td>
</tr>
<tr>
<td>48.0000</td>
<td>21.0000</td>
<td>10.14</td>
<td>96.14</td>
<td>3.86</td>
<td>1.7530</td>
</tr>
<tr>
<td>60.0000</td>
<td>4.0000</td>
<td>1.93</td>
<td>98.07</td>
<td>1.93</td>
<td>2.1913</td>
</tr>
<tr>
<td>72.0000</td>
<td>4.0000</td>
<td>1.93</td>
<td>100.00</td>
<td>0.00</td>
<td>2.6295</td>
</tr>
</tbody>
</table>

### Table BINPORT

<table>
<thead>
<tr>
<th>Entries in Table</th>
<th>Mean Argument</th>
<th>Standard Deviation</th>
<th>Sum of Arguments</th>
<th>Non-Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>108.0000</td>
<td>55.1893</td>
<td>22.7781</td>
<td>5960.4413</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upper Limit</th>
<th>Observed Frequency</th>
<th>Percent of Total</th>
<th>Percentage Cumulative</th>
<th>Cumulative Remainder of Mean</th>
<th>Multiple of Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.0000</td>
<td>4.0000</td>
<td>3.70</td>
<td>3.70</td>
<td>96.30</td>
<td>0.4349</td>
</tr>
<tr>
<td>36.0000</td>
<td>23.0000</td>
<td>21.30</td>
<td>25.00</td>
<td>75.00</td>
<td>0.6523</td>
</tr>
<tr>
<td>48.0000</td>
<td>22.0000</td>
<td>20.37</td>
<td>45.37</td>
<td>54.63</td>
<td>0.8697</td>
</tr>
<tr>
<td>60.0000</td>
<td>19.0000</td>
<td>17.59</td>
<td>62.96</td>
<td>37.04</td>
<td>1.0872</td>
</tr>
<tr>
<td>72.0000</td>
<td>17.0000</td>
<td>15.74</td>
<td>78.70</td>
<td>21.30</td>
<td>1.3046</td>
</tr>
<tr>
<td>84.0000</td>
<td>6.0000</td>
<td>5.56</td>
<td>84.26</td>
<td>15.74</td>
<td>1.5220</td>
</tr>
<tr>
<td>96.0000</td>
<td>10.0000</td>
<td>9.26</td>
<td>93.52</td>
<td>6.48</td>
<td>1.7395</td>
</tr>
<tr>
<td>108.0000</td>
<td>5.0000</td>
<td>4.63</td>
<td>98.15</td>
<td>1.85</td>
<td>1.9569</td>
</tr>
<tr>
<td>OVERFLOW</td>
<td>2.0000</td>
<td>1.85</td>
<td>100.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

**Average Value of Overflow is 116.2570**

### Table CINPORT

<table>
<thead>
<tr>
<th>Entries in Table</th>
<th>Mean Argument</th>
<th>Standard Deviation</th>
<th>Sum of Arguments</th>
<th>Non-Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>185.0000</td>
<td>31.2116</td>
<td>7.9542</td>
<td>5774.1465</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upper Limit</th>
<th>Observed Frequency</th>
<th>Percent of Total</th>
<th>Percentage Cumulative</th>
<th>Cumulative Remainder of Mean</th>
<th>Multiple of Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.0000</td>
<td>29.0000</td>
<td>15.68</td>
<td>15.68</td>
<td>84.32</td>
<td>0.7689</td>
</tr>
<tr>
<td>36.0000</td>
<td>119.0000</td>
<td>64.32</td>
<td>80.00</td>
<td>20.00</td>
<td>1.1534</td>
</tr>
<tr>
<td>48.0000</td>
<td>27.0000</td>
<td>14.59</td>
<td>94.59</td>
<td>5.41</td>
<td>1.5379</td>
</tr>
<tr>
<td>60.0000</td>
<td>9.0000</td>
<td>4.86</td>
<td>99.46</td>
<td>0.54</td>
<td>1.9224</td>
</tr>
<tr>
<td>72.0000</td>
<td>1.0000</td>
<td>0.54</td>
<td>100.00</td>
<td>0.00</td>
<td>2.3068</td>
</tr>
</tbody>
</table>
E-Functions

- In C-Functions and D-Functions, y values in the defining x,y-pairs are **numbers**

- In S-Functions, y values in the defining x,y-pairs are **identifiers** for members of a specified entity class

- There exists a class of Functions, E-Functions, whose y values in the defining x,y-pairs are **arithmetic expressions**
  (remember that constants and SNAs are degenerate cases of arithmetic expressions)

- Examples of E-Functions:

  **Example 1**
  
  \[
  \text{POINTER FUNCTION PF(INDEX),E3} \\
  \text{TYPE1,FN(SPEC1)/TYPE2,FN(SPEC2)/TYPE3,FN(SPEC3)}
  \]

  **Example 2**
  
  \[
  \text{RANDOM FUNCTION RN5,E4} \\
  .25,&\text{COLOR1/.50,&COLOR2/.75,&COLOR3/1.0,&COLOR4}
  \]

  **Example 3**
  
  \[
  \text{VARIED FUNCTION &KEY,E2} \\
  5,(3*PF(STEPTIME)+FN(LOWKEY))/10,RVEXPO(2,50)
  \]

- If a y-element in an E-Function involves one or more arithmetic operators, the y-element must be enclosed in parentheses (as in Example 3)
Model E5 is identical to Model E4
Except that the Type of Un/loading-Time Distribution is a function of the Type of Ship Involved

**E-Functions Are Used to Select the Correct Un/loading-Time Distribution**
(The y-values of E-Functions are Arithmetic Expressions)

Base Time Unit: 1 Hour

**Compiler Directives (EQU; SYM; UNLIST)**

<table>
<thead>
<tr>
<th>STYPE</th>
<th>EQU</th>
<th>1,PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASHIP</td>
<td>SYN</td>
<td>1, ASHIP is a synonym for 1</td>
</tr>
<tr>
<td>BSHIP</td>
<td>SYN</td>
<td>2, BSHIP is a synonym for 2</td>
</tr>
<tr>
<td>CSHIP</td>
<td>SYN</td>
<td>3, CSHIP is a synonym for 3</td>
</tr>
</tbody>
</table>

**UNLIST**

CSECHO suppresses Control-Statement echoes

**Control Statements (FUNCTION; QTABLE; STORAGE)**

**BRHTYPE FUNCTION**

PF(STYPE),S3,S

Berth-type identifier as a function of ship type

**CUNLOAD FUNCTION**

RN1,C6 0,18/.2,21/.4,25/.6,30/.8,32/1.0,33

Un/loading-time dist'n for CHIPS

**IMPORTQ FUNCTION**

PF(STYPE),S3,Q

In-port Queue identifier as a function of ship type

**STYPES FUNCTION**

RN1,D3 .45,ASHIP/.65,BSHIP/1,CSHIP

Distribution of types of ships

**TUGSIN FUNCTION**

PF(STYPE),D3

Number of berthing tugs needed as a function of ship type

**TUGSOUT FUNCTION**

PF(STYPE),D3

Number of deberthing tugs needed as a function of ship type

...un/loading time as a function of ship type...

**UNLODTRM FUNCTION**

PF(STYPE),E3

ASHIPS: normal distribution
BSHIPS: uniform 31 +/- 6 dist'n
CSHIPs: empirical distribution

**QTABLE**

ASHIPS, ABS(RVNORM(1,20,4))
BSHIP, ((31-6)+FRN1*2*6)
CSHIP, FN(CUNLOAD)

ASHIPS: distributions of in-port residence times for the three types of ships
* ABERTHS STORAGE 3 Type A berths
* BBERTHS STORAGE 2 Type B berths
* CBERTHS STORAGE 4 Type C berths
* TUGBOATS STORAGE 7 tugboats
*
* ******************************************************************************
* Model Segment 1 (Movement of Ships Through the Harbor)  *
* ******************************************************************************
*
* ...ships arrive...
GENERATE RVEXPO(1,3.9),,,,1PF
*
* ...let PF(STYPE) = type of ship
ASSIGN STYPE, FN(STYPES), PF
*
* ...start in-port Queue membership...
QUEUE FN(INPORTQ)
*
* ...request/capture berth-type needed...
ENTER FN(BRTHTYPE)
*
* ...request/capture tugboats needed for berthing...
ENTER TUGBOATS, FN(TUGSIN)
*
* ...berthing time...
ADVANCE 1.0
*
* ...give up the tugboats used for berthing...
LEAVE TUGBOATS, FN(TUGSIN)
*
* ...un/loading time...
ADVANCE FN(UNLODTYM)
*
* ...request/capture tugboats needed for deberthing...
ENTER TUGBOATS, FN(TUGOUT)
*
* ...deberthing time...
ADVANCE 0.5
*
* ...give up the tugboats used for deberthing...
LEAVE TUGBOATS, FN(TUGOUT)
*
* ...give up the berth...
LEAVE FN(BRTHTYPE)
*
* ...end in-port Queue membership...
DEPART FN(INPORTQ)
*
* ...steam off...
TERMINATE 1
*
******************************************************************************
* Run-Control Statements  *
******************************************************************************
*
START 100,NP  initialize with 100 ships;
RESET   reinitialize the model statistically
START   base steady-state results on

E-35