D.4.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
0 m2-heat-transfer-coefficient	opt-p		
1 m2-port-3b	p-obj		
2 m2-stream-3	p-obj		
3 m2-pipe	p-obj		
4 m2-edge-2a	p-obj		
5 n-Q-edge-1 b	p-obj		
6 m2-port-3a	p-obj		
7 m2-pipe-inside diameter	req-p		
8 m2-port-2 a	p-obj		
9 m2-port-1 b	p-obj		
10 m2-fluid-in-pipe	p-obj		
11 m2-stream-2	p-obj		
1 2 m2-stream-1	p-obj		
13 m2-mass-flow-rate	req-p		
14 m2-heat-capacity	W-P		
15 m2-density	W-P		
16 m2-thermal conductivity	req-p		
17 m2-viscosity	req-p		
18 m2-pipe-wall-temperature	opt-p		
19 m2-pipe-length	opt-p		
20 m2-pipe-axial direction	opt-p		
21 m2-outlet-temperature	opt-p		
22 m2-surface-normal-2 a	opt-p		
23 m2-inlet-temperature	opt-p		
24 m2-surface-normal-1 b	opt-p		
2 5 m2-neg-endpt-2a	neg-1		
26 m2-neg-endpt-1b	neg-2		
2 7 m2-neg-edge-3a	w - 3	intensive	(region-a region-y) (region-a region-x) (region-a region-c) (region-a region-d) (region-a region-e)
2 8 m2-neg-edge-2a	neg-4		

req-P -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
29 m2- neg s-normal-2 a	neg-5	intensive	(region -a region-b) (region -a region -c region-b) (region -a region -c region-d) (region -a region -c region-e) (region -a region d region-e) (region -a region d region-b) (region -a region -e region-b)
30 m2- neg -edge-l b	neg-6		
31 m2- neg -s-normal-l b	neg -7	intensive	(region -a region-c region-b region-d) (region -a region -c region-b region-e) (region -a region-c region-d region-e) (region -a region -d region-e region-b) (region -a region-c region-b region-d region-e)
32 m2-neg-port	neg-8		

req-P -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

D.5 Dittus-Boelter Model and PIPES-3 Equipment Description

D.5.1 Statistics on the Matching

Equipment Description:	PIPES-3
Model Description:	DITTUS-BOELTER
Goal:	calculate VALUES
Max. partial matches:	156
Complete Matches:	0
Rematches:	4
Run time:	0:11 (hr:min)
Objects created (kept):	104 (0)
Links created (kept):	557 (0)
Separately triggered reconfigurations:	
Used simple matching?	*Yes*
Used negative matching?	*Yes*
Used intensive reconfiguration?	*Yes*
Used part-whole reconfiguration?	*No*
Positive object triggered reconfiguration?	No
Parameter triggered reconfiguration?	*No*
Negative set triggered reconfiguration?	*Yes*

D.5.2 Generating the Potential Match Set

JNPUTS;
(SELECT- MODELS "wall-h-trans-coeff -c 'value 'describes 'pipe-3-port-3wc 'values)

OUTPUTS;
(
(wall- h-transcoeff -c value describes pipe -3-port-3wc values m2- heat -transfer-coeff icient
m2-port-3b)
)

D.5.3 Match/Reconfigure

INPUTS:
(DO-SIMPLE-MATCH 'm2-port-3b 'm2-heat-transfer-coefficient 'pipe-3-port-3wc
'wall-h-trans -coeff -c)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	**notes**	**Vector-list of matching Equipment Objects Returned**	
		vector #1	vector #2
		nil	

• different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-p -- required parameter opt-p -- optional parameter negative set matched

D.5.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS:
T

No conditions checked because no models matched.

D.5.5 Executing the Model

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS:
T

No conditions checked because no models matched.

D.5.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
0 m2- heat -transfer-coefficient	opt-p		
1 m2-port-3b	p-obj		
2 m2-stream-3	p-obj		
3 m2-pipe	p-obj		
4 m2-edge-2a	p-obj		
5 m2- edge-l b	p-obj		
6 m2-port-3a	p-obj		
7 m2- pipe -inside diameter	req-p		
8 m2-port-2a	p-obj		
9 m2-port-1b	p-obj		
10 m2-fluid-in-pipe	p-obj		
11 m2- stream -2	p-obj		
1 2 m2-stream-1	p-obj		
13 m2- mass -flow - rate	req-p		
14 m2- heat -capacity	W - P		
15 m2- density	W - P		
16 m2- thermal-conductivity	req-p		
17 m2- viscosity	req-p		
18 m2- pipe-wall -temperature	opt-p		
19 m2- pipe -length	opt-p		
20 m2- pipe -axial direction	opt-p		
21 m2-outlet -temperature	opt-p		
22 m2-surface-normal-2a	opt-p		
23 m2- inlet -temperature	opt-p		
2 4 m2-surface-normal-1 b	opt-p		
		intensive	(region -wc region -wd) (region -wc region -wb) (region -wc region -wd region -wb) (region -wc region -wd region -we) *termination*
25 m2-neg-endpt-2a	neg -1		

req-P -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

D.6 Dittus-Boelter Model and PIPES-4 Equipment Description

D.6.1 Statistics on the Matching

Equipment Description:	PIPES-4
Model Description:	DITTUS-BOELTER
Goal:	find EFFECTS
Max. partial matches:	a
Complete Matches:	0
Rematches:	3
Run time:	0:01 (hr:min)
Objects created (kept):	24 (0)
Links created (kept):	110 (0)
Separately triggered reconfigurations:	
Used simple matching?	*Yes*
Used negative matching?	*Yes*
Used intensive reconfiguration?	*Yes*
Used part-whole reconfiguration?	*No*
Positive object triggered reconfiguration?	*No*
Parameter triggered reconfiguration?	*Yes*
Negative set triggered reconfiguration?	*No*

D.6.2 Generating the Potential Match Set

INPUTS;
(SELECT -MODELS 'heat-capacity-b 'value 'describes 'region-b 'effects)

OUTPUTS;
(
(heat-capacity-b value describes region-b effects m2- heat -capacity m2-fluid -in -pipe)

D.6.3 Match/Reconfigure

INPUTS:
(DO-SIMPLE-MATCH 'm2-fluid-in-pipe 'm2-heat-capacity 'region-b 'heat-capacitiy-b)

OUTPUTS: (association-list and vector-list)

nil

D.6.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS:
T

No correct matches, so no conditions checked.

D.6.5 Executing the Model

INPUTS:
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS:
No correct matches, so no execution of model.

SIDF-EFFECTS: No sideeffects.

D.6.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconf igurations tried
0 m2- heat capacity	W - P		
1 m2- fluid -in -pipe	P - W		
2 m2- port-3 a	p-obj		
3 m2- port-2 a	p-obj		
4 m2- port-1 b	p-obj		
5 m2- density	req-P		
6 m2- thermal -conductivity	W - P		
7 m2- viscosity	W - P		
8 m2- stream3	p-obj		
9 m2- edge-1 b	p-obj		
10 m2- edge-2 a	p-obj		
11 m2- stream -2	p-obj		
12 m2- stream -1	p-obj	intensive	(region-b region -a) (region-b region c) (region -b region-c region-d) *termination*
13 m2- mass -flow - rate	req-p		

req-p -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

D.7 Dittus-Boelter Model and PIPES-5 Equipment Description

D.7.1 Statistics on the Matching

Equipment Description: PIPES-5
Model Description: DITTUS-BOELTER
Goal: find EFFECTS

Max. partial matches: 24
Complete Matches: 1
Rematches: 18
Run time: 0:14 (hr:min)
Objects created (kept): 147 (8)
Links created (kept): 1268 (29)
Separately triggered reconfigurations: 1

Used simple matching? Yes
Used negative matching? Yes
Used intensive reconfiguration? Yes
Used part-whole reconfiguration? No

Positive object triggered reconfiguration? Yes
Parameter triggered reconfiguration? No
Negative set triggered reconfiguration? No

D.7.2 Generating the Potential Match Set

INPUTS:
(SELECT-MODELS 'heat-capacity-c 'value 'describes 'region-c 'effects)

OUTPUTS;
(
(heat capacity-c value describes region -c effects m2- heat -capacity m2-fluid-in -pipe)
)

D.7.3 Match/Reconfigure

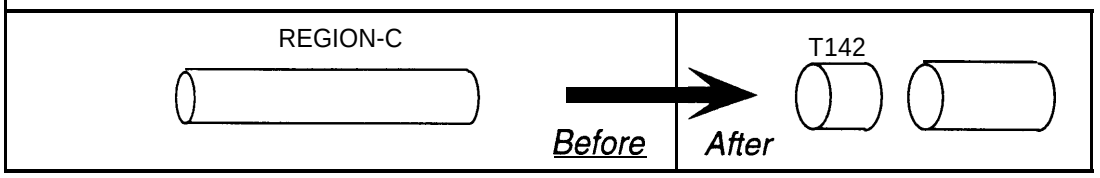
INPUTS:
(DC-SIMPLE-MATCH 'm2-fluid-in-pipe 'm2-heat-capacity 'region-c 'heat-capacity-c)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
0 m2-heat-capacity	req-P	heat-capacity-c	
1 m2-fluid-in-pipe	p-obj	t142	
2 m2-port-3a	p-obj	t152	
3 m2-port-1b	p-obj	t150	
4 m2-port-2a	p-obj	t138	
5 m2-density	req-p	rho-c	
6 m2-thermal-conductivity	req-p	thermal-cond-c	
7 m2-viscosity	req-p	mu-c	
8 m2-stream-3	p-obj	stream-pipe-2-c3	
9 m2-edge-1b	p-obj	t151	
10 m2-edge-2a	p-obj	t140	
11 m2-stream-1	p-obj	stream-bc	
12 m2-mass-flow-rate	W-P	inlet-flow-c	
13 m2-stream-2	p-obj	t139	
14 m2-port-3b	p-obj	pipe-2-port-3c	
15 m2-pipe	p-obj	pipe-2	
16 m2-pipe-inside diameter	W-P	pipe-2-id	
17 m2-inlet-temperature	opt-p	fluid-temp-port-cl	
18 m2-surface-normal-1b	opt-P	nil	
19 m2-outlet-temperature	opt-p	nil	
20 m2-surface-normal-2a	opt-P	nil	
21 m2-pipe-wall-temperature	opt-p	nil	
22 m2-heat-transfer-coefficient	opt-p	wall-h-trans-coeff-c	
23 m2-pipe-length	opt-p	nil	
24 m2-pipe-axial direction	opt-P	pipe-2-axial direction	
25 m2-neg-port	neg-1	nil	
26 m2-neg-edge-3a	neg-2	nil	
27 m2-neg-edge-1b	neg-3	nil	
28 m2-neg-s-normal-1b	neg 4	nil	
29 m2-neg edge-2 a	neg-5	nil	
30 m2-neg s-normal-2 a	neg 6	nil	
31 m2-neg-endpt-1b	neg-7	nil	
32 m2-neg-endpt-2a	neg-8	nil	

• different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matched

SIDE EFFECTS:



D.7.4 Checking Model Conditions

INPUTS:

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS:

T

((vectors-parallel?

m2- pipe -axial direction	value	m2- pipe -axial direction	value -type
m2- pipe -axial direction	value -interval-type	m2-pipe-axial-direction	value dimension
m2- pipe -axial direction	value -precision	m2- pipe -axial direction	value -precision-type
m2- pipe-axial direction	time	m2- pipe -axial direction	time-type
m2- pipe -axial direction	time -interval-type	m2- pipe -axial direction	time dimension
m2- pipe -axial direction	time -precision	m2- pipe -axial direction	time -precision-type
m2- surface -normal -1 b	value	m2-surface-normal-1 b	value -type
m2- surface -normal -1 b	value -interval-type	m2-surface-normal-1 b	value-dimension
m2- surface -normal -1 b	value- precision	m2- surface -normal -1 b	value -precision-type
m2- surface -normal-l b	time	m2- surface -normal-l b	time -type
m2- surface -normal -1 b	time -interval-type	m2-surface-normal-1 b	time dimension
m2- surface -normal -1 b	time -precision	m2- surface -normal -1 b	time -precision-type)

(vectors-parallel?

m2- pipe -axial direction	value	m2- pipe-axial direction	value -type
m2- pipe -axial direction	value -interval-type	m2- pipe -axial direction	value dimension
m2- pipe -axial direction	value -precision	m2- pipe -axial direction	value-precision-type
m2- pipe-axial direction	time	m2- pipe -axial direction	time -type
m2- pipe -axial direction	time -interval-type	m2- pipe-axial direction	time dimension
m2- pipe -axial direction	time -precision	m2- pipe -axial direction	time -precision-type
m2- surface -normal -2 a	value	m2- surface -normal -2 a	value -type
m2- surface -normal -2 a	value -interval-type	m2- surface -normal -2 a	value-dimension
m2- surface -normal -2 a	value- precision	m2- surface -normal -2 a	value -precision-type
m2- surface -normal-2 a	time	m2-surface-normal-2 a	time -type
m2- surface -normal -2a	time -interval-type	m2- surface -normal -2 a	time-dimension
m2- surface -normal -2a	time -precision	m2- surface -normal -2 a	time -precision-type)

(m2-dittus-boelter:pipe-temperatures-ok?

m2- inlet -temperature	value
m2-outlet -temperature	value
m2-pipe -wall-temperature	value)

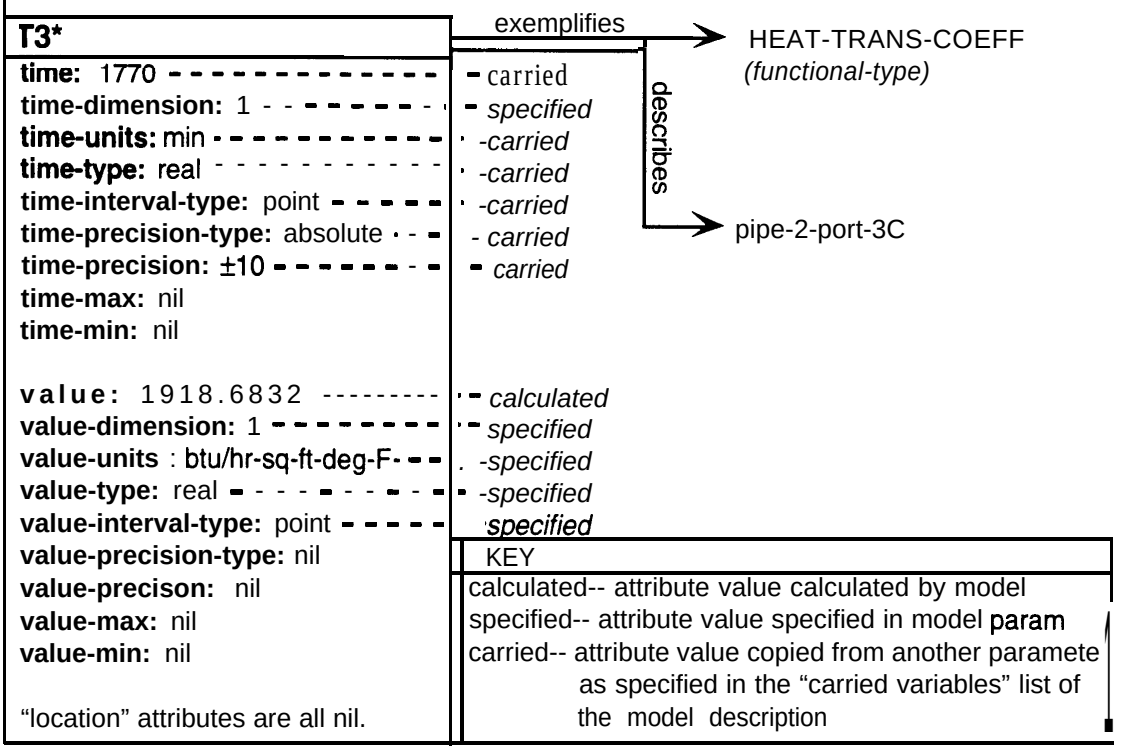
```
(m2-dittus-boelter:l-over-d-ok?
   m2- pipe -length      value          m2- pipe -inside diameter     value)

(any-common -portion?
   m2- viscosity                 time                    m2- viscosity                 time -type
   m2- viscosity                 time -interval-type     m2- viscosity                 time -precision
   m2- viscosity                 time -precision-type    n-&outlet-temperature         time
   m2-outlet -temperature        time -type              m2-outlet -temperature        time -interval-type
   m2-outlet -temperature        time -precision         m2-outlet -temperature        time -precision-type)

(any-common-portion?
   m2- viscosity                 time                    m2- viscosity                 time -type
   m2- viscosity                 time -interval-type     m2- viscosity                 time -precision
   m2- viscosity                 time-precision-type     m2- pipe -wall -temperature   time
   m2- pipe -wall -temperature   time -type              m2- pipe -wall -temperature   time -interval-type
   m2- pipe -wall -temperature   time -precision         m2- pipe-wall -temperature    time -precision-type)

(equal  m2-viscosity time -type      m2-pipe-wall-temperature time-type)

(equal  m2- viscosity time -type     m2-outlet -temperature time-type)

(equal  m2- viscosity time dimension     m2- pipe-wall-temperature time dimension)

(equal  m2- viscosity time dimension     m2-outlet-temperature time-dimension)

(equal  m2- viscosity time -units    m2- pipe -wall -temperature time-units)

(equal  m2-viscosity time-units      m2-outlet -temperature time-units))
```

D.7.5 Executing the Model

INPUTS:
(EXECUTE-MODEL assoc-list (first vector-list))

OUTPUTS:
(T3)

SIDE-EFFECTS: The parameter object, T3, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.7.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
0 m2-heat-capacity	req-p		
1 m2-fluid-in-pipe	p-obj		
2 m2-port-3a	p-obj		
3 m2-port-1 b	p-obj		
4 m2-port-2a	p-obj		
5 m2-density	req-p		
6 m2-thermal-conductivity	req-p		
7 m2-viscosity	W-P		
8 m2-stream-3	p-obj		
9 m2-edge-1 b	p-obj		
10 m2-edge-2a	p-obj		
11 m2-st ream-1	p-obj		
12 m2-mass-flow-rate	req-P		
13 m2-stream-2	p-obj	intensive--Note #1	(region-b region-c) (c-right)-- Note #2 (region-b c-left c-middle) (c-middle) (region-b c-left) (b-left) (b-right c-left) (b-right c-left c-middle) (c-middle c-right) (b-right region-c) (region-c region-d) (c-left c-middle) (c-left)
14 m2-port-3b	p-obj		
15 m2-pipe	p-obj		
16 m2-pipe-inside diameter	req-p		
17 m2-inlet-temperature	opt-p		
18 m2-surface-normal-1 b	opt-p		
19 m2-outlet-temperature	opt-p		
20 m2-surface-normal-2 a	opt-p		
21 m2-pipe-wall-temperature	opt-p		
22 m2-heat-transfer-coeff icient	opt-p		

req-P -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
23 m2- pipe -length	opt-p		
24 m2- pipe -axial direction	opt-p		
25 m2- neg -port	neg -1		
26 m2- neg -edge-3 a	neg -2		
27 m2-neg-edge-1 b	neg-3		
28 m2- neg- s-normal -1 b	neg-4		
29 m2- neg edge-2 a	neg-5		
30 m2- neg -s- normal-2 a	neg-6		
31 m2-neg-endpt-1 b	neg -7		
3 2 m2-neg-endpt-2 a	neg-8		

req-P -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

Note #1: This intensive reconfiguration is not triggered by inability to find an equipment object to match M2-PORT-3B, but by the attempt to find a "has-edge" link from objects matching M2-PORT-2A (4) to objects matching M2-EDGE-2A (10). At this point the matcher was finding all objects linked to M2-PORT-2A. It successfully found and matched M2-STREAM-2, but could not correctly match the "has-edge" link to the objects matching M2-EDGE-2A, which was already found as edge of the port M2-PORT-3A. The cylindrical surface of the liquid region must share edges with the ends of the region, and this condition is not met in the current liquid region, Region-C.

Note #2: This case does both merges and splits. The split regions are referred to as "x-left", x-middle", "x-right", where x is the letter naming the original region. Region-B can be split along the boundary of Pipe-l and Pipe-2, with the two halves called "B-left" (portion in Pipe-l) and "B-right" for the portion in Pipe-2. Region-C can be split into 3 portions corresponding to the liquid in Pipe-2 ("C-left"), Pipe-3 ("C-middle"), and Pipe-4 ("C-right").

D.8 Dittus-Boelter Model and PIPES-6 Equipment Description

D.8.1 Statistics on the Matching

Equipment Description: PIPES-6
Model Description: DITTUS-BOELTER
Goal: find EFFECTS

Max. partial matches: 40
Complete Matches: 0
Rematches: 20
Run time: 0:15 (hr:min)
Objects created (kept): 183 (0)
Links created (kept): 1230 (0)
Separately triggered reconfigurations: 2

Used simple matching? Yes
Used negative matching? Yes
Used intensive reconfiguration? Yes
Used part-whole reconfiguration? No

Positive object triggered reconfiguration? Yes
Parameter triggered reconfiguration? No
Negative set triggered reconfiguration? No

D.8.2 Generating the Potential Match Set

INPUTS;
(SELECT-MODELS 'thermal-cond-b 'value 'describes 'region-b 'effects)

OUTPUTS;
(
(thermal-cond-b value describes region-b effects m2-thermal-conductivity m2-fluid-in-pipe)
)

D.8.3 Match/Reconfigure

INPUTS:
(DO -SIMPLE-MATCH 'm2-fluid-in-pipe 'm2-thermal-conductivity 'region-B 'thermal-cond-b)

OUTPUTS: (association-list and vector-list)

nil

D.8.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS;
T

No conditions checked because no matches found.

D.8.5 Executing the Model

INPUTS:
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS:
No model executed because no matches found.

SIDE-EFFECTS: No side effects..

D.8.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)		notes	Reconfiguration type	Reconfigurations tried
0	m2- thermal -conductivity	req-P		
1	m2-fluid-in -pipe	p-obj		
2	m2-port-3a	p-obj		
3	m2-port-1b	p-obj		
4	m2- port-2 a	p-obj		
5	m2- heat -capacity	W - P		
6	m2- density	req-p		
7	m2- viscosity	req-P		
8	m2- stream3	P-W		
9	m2-edge-1b	P-W		
10	m2- edge-2 a	p-obj		
11	m2- stream -1	p-obj		
12	m2- mass -flow-rate	req-P		

req-p -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
13 m2- stream -2	p-obj	intensive--Note #1	(region-b region-a) (b-4) -- Note #2 (region-a bl b-2 b-3) (b-3) (region-a bl b-2) (region-a b-1) (b-2) (b-2 b3) (b-3 b4) (b-2 b3 b-4) (region-b region-c) (b-l b-2 b-3) (b-4 region-c) (b-l b2) (b-3 b-4 region-c) (b-2 b3 b-4 region-c) (b-1)
14 m2-port-3b	p-obj	intensive	*termination* --Note #3
15 m2- pipe	p-obj		

req-P -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

Note #1: This intensive reconfiguration is not triggered by inability to find an equipment object to match M2-PORT-3B, but by the attempt to find a "has-edge" link from objects matching M2-PORT-2A (4) to objects matching M2-EDGE-2A (10). At this point the matcher was finding all objects linked to M2-PORT-2A. It successfully found and matched M2-STREAM-2, but could not correctly match the "has-edge" link to the objects matching M2-EDGE-2A, which was already found as edge of the port M2-PORT-3A. The cylindrical surface of the liquid region must share edges with the ends of the region, and this condition is not met in the current liquid region, Region-C.

Note #2: This case does both merges and splits. The split regions are referred to as "B-l ", "B-2", "B-3", and "B-4" where B is the letter naming the original region. Region-B can be split along the boundary of Elbow-l, Pipe-2, Pipe-3, and Pipe-4, where the pieces are numbered from left to right, with B-l corresponding to the liquid region inside Elbow-l.

Note #3: No further merges or splits were available, because all regions had previously been tested and failed to rematch the objects matched so far. Thus the intensive reconfiguration heuristic prevents any further merges or splits from being tested.

D.9 Dalle-Molle Model and CSTR-1 Equipment Description

D.9.1 Statistics on the Matching

Equipment Description:	CSTR-1	
Model Description:	DALLE-MOLLE	
Goal:	calculate VALUES	
Max. partial matches:	240	
Complete Matches:	16	
Rematches:	0	
Run time:	0:13	(hr:min)
Objects created (kept):	1 (1)	
Links created (kept):	2 (2)	
Separately triggered reconfigurations:	0	
Used simple matching?	Yes	
Used negative matching?	Yes	
Used intensive reconfiguration?	No	
Used part-whole reconfiguration?	No	
Positive object triggered reconfiguration?	No	
Parameter triggered reconfiguration?	No	
Negative set triggered reconfiguration?	No	

D.9.2 Generating the Potential Match Set

INPUTS;
(SELECT-MODELS 'outlet-temperature-2771 'value 'describes 'port-liquid-7 'values)

OUTPUTS:
(
(outlet -temperature -2771 value describes port-liquid-7 values m8- tempout-after m&port-liq outlet)
)

D.9.3 Match/Reconfigure

INPUTS:
(DO-SIMPLE-MATCH
'm8-port-liq-outlet 'm8-tempout-after 'port-liquid-7 'outlet-temperature-2771)
OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned (See NOTE)	
		vector #1	vector #5
0 m8- tempout-after	W-P	outlet- temperature -2771	outlet-temperature -2771
1 m8- port- liq -outlet	p-obj	port-liquid-7	port- liquid-7
2 m&edge-outlet	p-obj	edge- liquid-9	edge- liquid-9
3 m8- tempout- before	req-P	outlet-temperature-2766	outlet- temperature -2766
4 m&liquid	p-obj	r12- liquid	r12- liquid
5 m8- port- nozzle -liquid	p-obj	port-onoz-4	port-onoz-4
6 m8-port-liq-1-inlet	p-obj	port-liquid3	port- liquid-l .
7 m8- port- liq -2-inlet	p-obj	port-liquid-l	port-liquid-3*
8 m8- port- liq -impeller	p-obj	port- liquid6	port- liquid6
9 m8- height- before	req-P	liquid- height-2766	liquid- height-2766
10 m&outlet -nozzle	p-obj	r12-outlet-nozzle	r12-outlet-nozzle
1 1 m8-coldflow-after	req-P	hot -inlet -flow -2771	cold-inlet-flow -2771*
12 m8-coldflow- before	req-P	hot -inlet -flow -2766	cold- inlet-flow -2766*
13 m8-hotflow -after	W-P	cold- inlet-flow -2771	hot -inlet -flow -2771*
14 m8-hotflow -before	req-P	cold- inlet-flow -2766	hot -inlet-flow-2766*
15 m&stream -liq -impeller	p-obj	stream -imp-liq	stream -imp-liq
16 m8-hot-cold-mixer	p-obj	r12-cstr	r12- cstr
17 m8- port- nozzle -floor	p-obj	port- onoz-2	port-onoz-2
18 m&port- impeller-liquid	p-obj	port- imp-l	port- imp-l
1 9 m&tank-floor	p-obj	r12-tank-floor	r12-tank-floor
20 m8- impeller	P-W	r12- impeller	r12- impeller
21 m&stream-floor-outlet	p-obj	stream -tfloor-onoz	stream -tfloor-onoz
22 m8- port-floor-outnoz	p-obj	port-tfloor-4	port-tfloor-4
23 m&outflow -after	opt-P	nil	nil
24 m&outflow -before	opt-p	nil	nil
25 m8- height-deriv- before	opt-p	nil	nil
26 m&height-after	opt-p	nil	nil
27 m8- cflow deriv -after	opt-p	nil	nil
28 m8-cflow deriv-before	opt-p	nil	nil
29 m8-ctemp-deriv- before	opt-p	nil	nil
30 m8-ctemp-deriv-after	opt-p	nil	nil
31 m8-coldtemp-after	opt-p	nil	nil
32 m8- coldtemp -before	opt-p	nil	nil
33 m8- hflow-deriv-before	opt-p	nil	nil
34 m8- htemp deriv -after	opt-p	nil	nil
35 m8- hflowderiv-after	opt-p	nil	nil
36 m8- htemp deriv -before	opt-p	nil	nil
37 m8- hottemp-after	opt-P	nil	nil
38 m8- hottemp- before	opt-p	nil	nil
39 m8- neq -port-liquid	neg-1	nil	nil

OUTPUTS: -CONTINUED

Index from Assoc-list	Vector-list of matching Equipment Objects Returned	
	vector #9	vector #13
0	outlet-temperature -2771	outlet-temperature-2771
1	port- liquid-7	port-liquid-7
2	edge-liquid-9	edge- liquid-9
3	outlet-temperature-2766	outlet-temperature-2766
4	r12- liquid	r12- liquid
5	port- onoz-1 *	port-onoz-l*
6	port- liquid3	port- liquid-l *
7	port-liquid-l	port-liquid-3*
8	port- liquid-6	port- liquid6
9	liquid- height-2766	liquid- height-2766
10	r12-outlet-nozzle	r12- outlet-nozzle
11	hot -inlet -flow-2771	cold- inlet-flow -2771 .
12	hot -inlet -flow -2766	cold-inlet-flow-2766*
13	cold-inlet-flow-2771	hot- inlet-flow -2771 .
14	cold-inlet -flow-2766	hot -inlet -flow -2766*
15	stream -imp-liq	stream -imp-liq
16	r12-cst r	r12-cstr
17	port- onoz-2	port- onoz-2
18	port- imp-l	port- imp-l
19	r12- tank-floor	r12-tank-floor
20	r12- impeller	r12- impeller
21	stream -tfloor-onoz	stream-tfloor-onoz
22	port-tfloor-4	port-tfloor-4
23	nil	nil
24	nil	nil
25	nil	nil
26	nil	nil
27	nil	nil
28	nil	nil
29	nil	nil
30	nil	nil
31	nil	nil
32	nil	nil
33	nil	nil
34	nil	nil
35	nil	nil
36	nil	nil
37	nil	nil
38	nil	nil
39	nil	nil

* different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matchec

NOTE: 16 vectors (correct matches) were returned, but only the 1 st, 5th, 9th, and 13th are shown here because all the others do not meet the approximation conditions. (The function CHECK-MODEL-CONDITIONS returns NIL when called on those other vectors.) These 16 different vectors result from multiple ways in which some objects and parameters can be matched. At the matching stage, the relative times of parameters are not considered, so M8-COLDFLOW-BEFORE (a model parameter) can be matched to either the initial cold flow rate or the later cold flow rate (2 possibilities). Similarly M8-HOTFLOW-BEFORE can be matched to either of the two available flow rates (2 possibiliites). The matching stage also does not distinguish between relative values of parameters, so it can match M8-PORT-LIQ-1-INLET, the inlet for the cold flow, to either the cold inlet or the hot inlet (2 possibilities). This model description also requires that the outlet port of the liquid share an edge with the outlet nozzle, and the outlet nozzle actually has two ports that have such an edge (2 possibilities). There are 16 (2 X 2 X 2 X 2) combinations.

D.9.4 Checking Model Conditions

INPUTS;
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 0 vector-list))

OUTPUTS:
T

((m8-dalle-molle::flow-proportional-to-height?
m&outflow -before value m8- outflow -after value
m8- outflow -after value -precision-type m8- height- before value
m8- height-after value m&height-after value -precision-type)

(equal m8-hottemp-before value m8- hottemp-after value)

(equal m8-coldtemp-before value m8-coldtemp-after value)

(equal m8-htemp-deriv-after value nil 0)

(equal m8-htemp-deriv- before value nil 0)

(equal m8-ctemp-deriv-after value nil 0)

(equal m8-ctemp-deriv-before value nil 0)

(equal m8-hflow-deriv-after value nil 0)

(equal m8-hflow-deriv-before value nil 0)

(equal m&height -deriv-before value nil 0)

(equal m8-cflow-deriv-after value nil 0)

(equal m8-cflow-deriv-before value nil 0)

(m8-dalle-molle:: qspaces-have-common-element?
m8- coldtemp -before value -precision-type nil 2
m8- tempout- before value -precision-type nil 2)

(m8- dalle -molle:: qspaces-have -common element ?
m8- hottemp- before value -precision-type nil 2
m8- tempout- before value -precision-type nil 4)

(possibly-equal?
m8-coldflow-after time m8-coldflow-after time-type
m8-coldflow-after time-interval-type m8-coldflow-after time -precision
m8-coldflow-after time-precision-type m&outflow-after time
m8- outflow -after time -type m8- outflow -after time-interval-type
m&outflow -after time -precision m8- outflow -after time -precision-type)

(possibly-equal?
mkoldflow-after time m8- coldflow-after time -type
m8-coldflow-after time -interval-type mkoldflow-after time -precision
m8-coldflow-after time -precision-type m8- htemp -deriv-after time
m8- htemp -deriv-after time -type m8- htemp clerk-after time -interval-type
m8- htemp -deriv-after time -precision m8- htemp deriv-after time -precision-type)

OUTPUTS:-continued

(possibly equal?

m8-coldflow-after	time	m8-coldflow-after	time -type
mkoldflow-after	time -interval-type	mkoldflow-after	time -precision
mkoldflow-after	time-precision-type	m8-ctemp-deriv-after	time
m8-ctemp-deriv-after	time-type	m8-ctemp-deriv-after	time-interval-type
m8-ctemp-deriv-after	time-precision	m8-ctemp-deriv-after	time-precision-type)

(possibly-equal?

mkoldflow-after	time	mkoldflow-after	time-type
m8-coldflow-after	time -interval-type	m8- coldflow-after	time -precision
mkoldflow-after	time-precision-type	m8- hflowderiv-after	time
m8- hflow deriv -after	time -type	m8- hflowderiv-after	time-interval-type
m8- hflow deriv -after	time -precision	m8- hflowderiv-after	time-precision-type)

(possibly-equal?

m8-coldflow-after	time	m8-coldflow-after	time-type
m8-coldflow-after	time -interval-type	mkoldflow-after	time -precision
m8-coldflow-after	time-precision-type	m8-cflow-deriv-after	time
m8-cflow-deriv-after	time-type	m8-cflow-deriv-after	time-interval-type
m8-cflow-deriv-after	time -precision	m8-cflow-deriv-after	time-precision-type)

(possibly equal?

m8-coldflow- before	time	m8-coldflow-before	time-type
m8-coldflow-before	time-interval-type	m8- coldf low- before	time -precision
m8-coldflow- before	time -precision-type	m&outflow -before	time
m&outflow -before	time -type	m8- outflow -before	time -interval-type
m&outflow -before	time -precision	m&outflow -before	time -precision-type)

(possibly-equal?

m8- coldf low- before	time	m8-coldflow- before	time-type
m8-coldflow- before	time -interval-type	m8-coldflow- before	time -precision
m8-coldflow- before	time -precision-type	m8- htempderiv-before	time
m8- htemp clerk-before	time -type	m8- htemp deriv -before	time -interval-type
m8- htemp clerk-before	time -precision	m8- htempderiv-before	time -precision-type)

(possibly-equal?

m8-coldflow- before	time	m8-coldflow- before	time -type
m8- coldflow- before	time -interval-type	m8-coldflow- before	time -precision
m8-coldflow- before	time-precision-type	m8-ctemp-deriv-before	time
m8-ctemp-deriv-before	time-type	m8-ctemp-deriv- before	time-interval-type
. m8-ctemp-deriv-before	time-precision	m8-ctemp-deriv- before	time-precision-type)

(possibly equal?

m8-coldflow-before	t i m e	m8-coldflow- before	time-type
m8-coldflow- before	time-interval-type	m8- coldf low- before	time -precision
m8-coldflow- before	time -precision-type	m8- hflow -deriv-before	time
m8- hflow deriv-before	time -type	m8- hflow -deriv-before	time -interval-type
m8- hflow deriv-before	time -precision	m8- hflow clerk-before	time -precision-type)

OUTPUT&-continued

(**possibly-equal?**
m8-coldflow- before time m8-coldflow- before time -type
m8-coldflow- before time -interval-type m8-coldflow- before time -precision
m8- coldf low- before time -precision-type m8- height-deriv- before time
m8- height-deriv- before time-type m8-height-deriv- before time -interval-type
m8- height-deriv- before time -precision m8- height-deriv- before time -precision-type)

(**possibly-equal?**
m8-coldflow- before time m8- coldflow- before time -type
m8- coldflow- before time -interval-type m8-coldflow- before time -precision
m8-coldflow- before time -precision-type m8- cflow deriv -before time
m8-cflow deriv -before time -type m8-cflow-deriv-before time-interval-type
m8-cflow-deriv-before time-precision m8-cflow-deriv-before time-precision-type)

(possibly-before?
m8-coldtemp -before value m8-coldtemp -before value -type
m8- coldtemp -before value -interval-type m8-coldtemp -before value -precision
m8- coldtemp -before value -precision-type m8- hottemp- before value
m8- hottemp- before value -type m8- hottemp- before value -interval-type
m8- hottemp- before value -precision m8- hottemp- before value -precision-type)

(equal m&height-after time-units **m8-coldflow-before** time-units)

(equal m8- htempderiv-after time-units **m8-coldflow-before** time-units)

(equal m8- htemp deriv -before time-units mkoldflow-before time -units)

(equal m8-ctemp-deriv-after time-units **m8-coldflow-before** time-units)

(equal m8-ctemp-deriv- before time-units **m8-coldflow-** before time -units)

(equal m8- hflowderiv-after time-units mkoldflow-before time-units)

(equal m8- hflow deriv -before time -units **m8-coldflow-** before time -units)

(equal m8- height-deriv- before time-units **m8-coldflow-** before time -units)

(equal m8-cflow-deriv-after time-units **m8-coldflow-before** time-units)

(equal m8-cflow-deriv-before time-units **m8-coldflow-before** time-units))

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 4 vector-list)) ← 5th vector
T
[The list of unverified conditions returned is the same as for the first vector.]

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 8 vector-list)) ← 9th vector
T
[The list of unverified conditions returned is the same as for the first vector.]

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 12 vector-list)) ← 13th vector
T
[The list of unverified conditions returned is the same as for the first vector.]

D.9.5 Executing the Model

INPUTS:
(EXECUTE-MODEL assoc-list (first vector-list))

OUTPUTS:

[NOTE: All of the following is printed to the screen while the QSIM system is being loaded.]

; Loading SYS: SITE; NQ-92.TRANSLATIONS#> into package FILE-SYSTEM
; Loading SYS: SITE; NQ-92.SYSTEM#> into package USER
; Loading SYS: SITE; NQ-92.TRANSLATIONS#> into package FILE-SYSTEM
Current login name is MURDOCK for host HPP.Stanford.EDU.
Type either password or loginname<space>password: XXXXXXXmurdock XXXXXXXX

File to Read

NQ-92: NQ-92; TI-NQ-SYSTEM.LISP#>

Read it? (Y or N) Yes.
; Reading NQ-92: NQ-92; TI-NQ-SYSTEM.LISP#>
; Loading SYS: SITE; XPOS-92.TRANSLATIONS#> into package FILE-SYSTEM
; Loading SYS: SITE; XPOS-92.SYSTEM#> into package USER
; Loading SYS: SITE; XPOS-92.TRANSLATIONS#> into package FILE-SYSTEM

; Reading Xpos-92: XPOS-92; XPOS-SYSTEM.LISP#>
Loading Xpos-92: XPOS-BIN; POSTSCRIPTER.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; QPLOT-PRIMITIVES.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; QPLOT-ADVANCED.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; PPLOT.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; TI-MACHINE-PARAMS.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; TI-POS-USRINTFACE.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; QGRAPH.XLD#> into package POS
Loading Xpos-92: XPOS-BIN; QGRAPH-IO.XLD#> into package POS
Making XPOS-92 patchable
Experimental XPOS version 1. loaded

If you use the :defaulted-batch option, compiler warnings for
system NQ-92 will be written to the file nq-2:nq-92;cwarns.lisp.

Files to be loaded:

NQ-92: NQ-BIN; QDEFS.XLD#>
NQ-92: NQ-BIN; PLOTTING-CONSTANTS.XLD#>
NQ-92: NQ-BIN; QUTILS.XLD#>
NQ-92: NQ-BIN; DYNAMIC-SLOTS.XLD#>
NQ-92: NQ-BIN; STRUCTURES.XLD#>
NQ-92: NQ-BIN; DEFOTHERS.XLD#>
NQ-92: NQ-BIN; PROLOG.XLD#>

OUTPUTS:

NQ-92: NQ-BIN; **DEFINE-QDE.XLD#>**
NQ-92: NQ-BIN; **CFILTER.XLD#>**
NQ-92: NQ-BIN; **CONSTRAINTS.XLD#>**
NQ-92: NQ-BIN; **QSPACE-HIERARCHY.XLD#>**
NQ-92: NQ-BIN; **PROPAGATION.XLD#>**
NQ-92: NQ-BIN; **STATES.XLD#>**
NQ-92: NQ-BIN; **GLOBAL-FILTERS.XLD#>**
NQ-92: NQ-BIN; **TRANSITIONS.XLD#>**
NQ-92: NQ-BIN; QSIM-TO-QGRAPH.XLD#>
NQ-92: NQ-BIN; **QPLOT.XLD#>**
NQ-92: NQ-BIN; **ALLOCATIONS.XLD#>**
NQ-92: NQ-BIN; TIME-PLOT.XLD#>
NQ-92: NQ-BIN; NUMERIC-PLOT.XLD#>
NQ-92: NQ-BIN; **EXAMINE.XLD#>**
NQ-92: NQ-BIN; SPECIAL-DISPLAYS.XLD#>
NQ-92: NQ-BIN; **VERSIPLOT.XLD#>**
NQ-92: NQ-BIN; **TABLES.XLD#>**
NQ-92: NQ-BIN; FRONT-END.XLD#>
NQ-92: NQ-BIN; **SYNTAX.XLD#>**
NQ-92: NQ-BIN; HQ-PLOT.XLD#>
NQ-92: NQ-BIN; **VIEWERS-CLISP-2.XLD#>**
NQ-92: NQ-BIN; VIEWERS-MISC.XLD#>
NQ-92: NQ-BIN; VIEWERS-CLISP.XLD#>
NQ-92: NQ-BIN; **EQUATIONS.XLD#>**
NQ-92: NQ-BIN; **RK45S.XLD#>**
NQ-92: NQ-BIN; TIME-SCALE.XLD#>
NQ-92: NQ-BIN; HOD-DERIVATION.XLD#>
NQ-92: NQ-BIN; **NCURVATURE.XLD#>**
NQ-92: NQ-BIN; **QMATCH.XLD#>**
NQ-92: NQ-BIN; HOD-SYSTEM-PROPERTY.XLD#>
NQ-92: NQ-BIN; **ARITHMETIC.XLD#>**
NQ-92: NQ-BIN; **Q2.XLD#>**
NQ-92: NQ-BIN; **NIC.XLD#>**
NQ-92: NQ-BIN; **ENERGY-GF.XLD#>**
NQ-92: NQ-BIN; **COMPARISON.XLD#>**
NQ-92: NQ-BIN; OCC-BRANCH-ELIMINATION.XLD#>
NQ-92: NQ-BIN; FOCUSING-TECHNIQUES.XLD#>
NQ-92: NQ-BIN; NUMERIC-EQN-GEN.XLD#>
NQ-92: NQ-BIN; **NSIM-INTERVAL.XLD#>**
NQ-92: NQ-BIN; **NSIM-INIT.XLD#>**
NQ-92: NQ-BIN; **XEDGE.XLD#>**
NQ-92: NQ-BIN; ENVISION-GUIDE-FILTER.XLD#>

OUTPUTS;
Files to be Read:

NQ-92: CATALOGS; EXAMPLES.LISP#>
NQ-92: CATALOGS; TSA.LISP#>
NQ-92: CATALOGS; EXTRAS.LISP#>
NQ-92: CATALOGS; CHATTER4.LISP#>
NQ-92: CATALOGS; CHATTER3.LISP#>
NQ-92: CATALOGS; CHATTER2.LISP#>
NQ-92: CATALOGS; CHATTER1.LISP#>
NQ-92: CATALOGS; Q-FEATURES.LISP#>
NQ-92: CC; CC-CATALOG.LISP#>

Files to be loaded:

NQ-92: NQ-BIN; DEFINITIONS.XLD#>
NQ-92: NQ-BIN; FUNCTIONS.XLD#>
NQ-92: NQ-BIN; QSIM-EXPORTS.XLD#>

Files to be Read:

NQ-92: CC; HYDRAULIC.LISP#>
NQ-92: CC; ELECTRICAL.LISP#>
NQ-92: NQ-92; TI-USER-INTERFACE.LISP#>

Going to Make NQ-92: The New QSIM patchable

Load or Read all sixty-four of them and Make NQ-92: The New QSIM
patchable? (S, Y, or N) Yes.
Loading NQ-92: NQ-BIN; QDEFS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; PLOTTING-CONSTANTS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; QUTILS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; DYNAMIC-SLOTS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; STRUCTURES.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; DEFOTHERS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; PROLOG.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; DEFINE-QDE.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; CFILTER.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; CONSTRAINTS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; QSPACE-HIERARCHY.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; PROPAGATION.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; STATES.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; GLOBAL-FILTERS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; TRANSITIONS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; QSIM-TO-QGRAPH.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; QPLOT.XLD#> into package QSIM

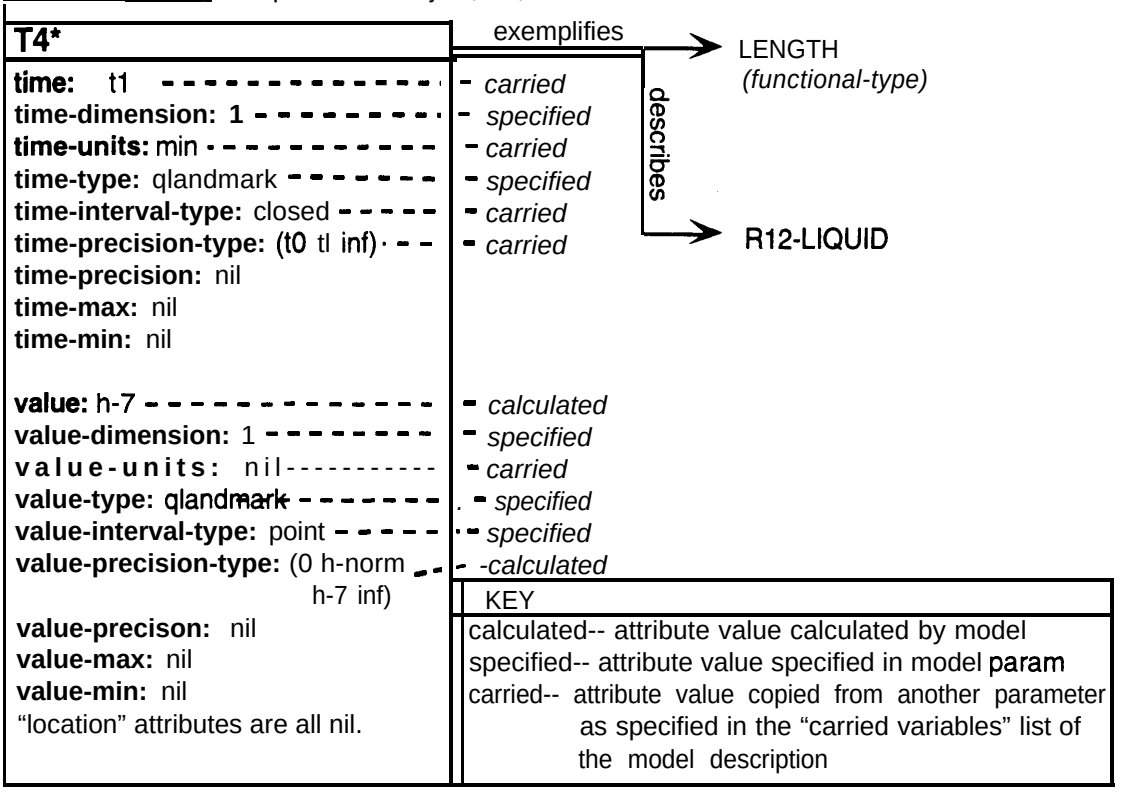
OUTPUTS:

Loading NQ-92: NQ-BIN; ALLOCATIONS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; TIME-PLOT.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; NUMERIC-PLOT.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; EXAMINE.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; SPECIAL-DISPLAYS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; VERSIPLOT.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; TABLES.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; FRONT-END.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; SYNTAX.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; HQ-PLOT.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; VIEWERS-CLISP-2.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; VIEWERS-MISC.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; VIEWERS-CLISP.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; EQUATIONS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; RK45S.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; TIME-SCALE.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; HOD-DERIVATION.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; NCURVATURE.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; QMATCH.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; HOD-SYSTEM-PROPERTY.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; ARITHMETIC.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; Q2.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; NIC.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; ENERGY-GF.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; COMPARISON.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; OCC-BRANCH-ELIMINATION.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; FOCUSING-TECHNIQUES.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; NUMERIC-EQN-GEN.XLD#b into package QSIM
Loading NQ-92: NQ-BIN; NSIM-INTERVAL.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; NSIM-INIT.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; XEDGE.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; ENVISION-GUIDE-FILTER.XLD#> into package QSIM
Reading NQ-92: CATALOGS; EXAMPLES.LISP#> into package QSIM
Reading NQ-92: CATALOGS; TSA.LISP#> into package QSIM
Reading NQ-92: CATALOGS; EXTRAS.LISP#> into package QSIM
Reading NQ-92: CATALOGS; CHATTER4.LISP#> into package QSIM
Reading NQ-92: CATALOGS; CHATTER3.LISP#> into package QSIM
Reading NQ-92: CATALOGS; CHATTER2.LISP#> into package QSIM
Reading NQ-92: CATALOGS; CHATTER1 .LISP#> into package QSIM
Reading NQ-92: CATALOGS; Q-FEATURES.LISP#> into package QSIM
Reading NQ-92: CC; CC-CATALOG.LISP#> into package QSIM
Loading NQ-92: NQ-BIN; DEFINITIONS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; FUNCTIONS.XLD#> into package QSIM
Loading NQ-92: NQ-BIN; QSIM-EXPORTS.XLD#> into package QSIM
Reading NQ-92: CC; HYDRAULIC.LISP#> into package QSIM
Reading NQ-92: CC; ELECTRICAL.LISP#> into package QSIM
Reading NQ-92: NQ-92; TI-USER-INTERFACE.LISP#> into package QSIM

OUTPUTS;

```
• ** Type <system>-q to use the QSIM User Interface ***
Making NQ-92: The New QSIM patchable
Experimental NQ-92 version 4. loaded
; Loading KSL-EXP-23: MURDOCK.MODELS; DALLE-MOLLE-INTERFACE.LISP#>
      into package QSIM

Run time: 0.283 seconds to initialize a state.
Run time: 0.083 seconds to initialize a state.
Run time: 0.800 seconds to simulate 5 states.
 Final states are: (S-1 1)

(OUTLET-TEMPERATURE-2771  T4)     [This is the list returned by executing the model]
```

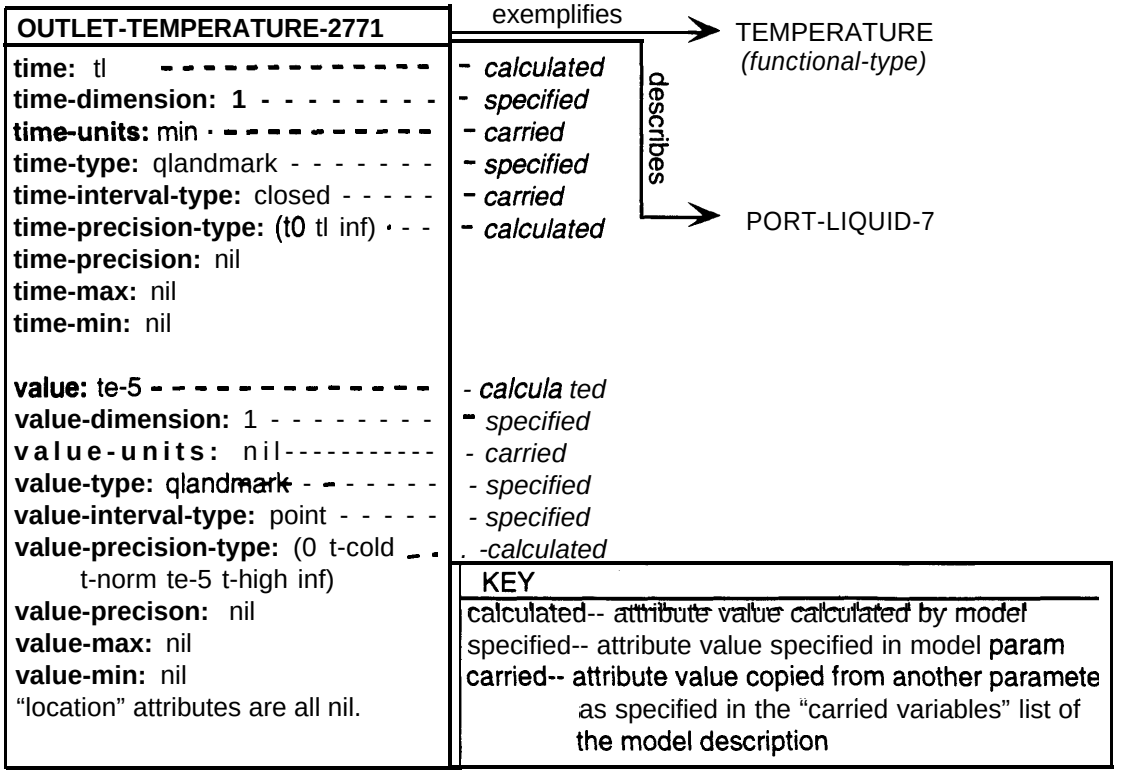
SIDE-EFFECTS; The parameter object, T4, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

SIDE-EFFECTS: --continued

The parameter object, OUTLET-TEMPERATURE-2771, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.9.6 Summary of Merges and Splits Tested

D.10 Dalle-Molle Model and CSTR-2 Equipment Description

D.10.1 Statistics on the Matching

Equipment Description:	CSTR-2
Model Description:	DALLE-MOLLE
Goal:	calculate VALUES
Max. partial matches:	1840
Complete Matches:	16
Rematches:	33
Run time:	4:46 (hr:min)
Objects created (kept):	? (77)
Links created (kept):	? (334)
Separately triggered reconfigurations:	3
Used simple matching?	Yes
Used negative matching?	Yes
Used intensive reconfiguration?	Yes
Used part-whole reconfiguration?	Yes
Positive object triggered reconfiguration?	Yes
Parameter triggered reconfiguration?	Yes
Negative set triggered reconfiguration?	Yes

D.10.2 Generating the Potential Match Set

INPUTS:
(SELECT-MODELS 'outlet-temperature-2771 'value 'describes 'port-liq-d-6 'values)

OUTPUTS;
(
(outlet-temperature-2771 value describes port-liqd-6 values m8-tempout-after m8- port-liq -outlet)

D.10.3 Match/Reconfigure

INPUTS:
(DO-SIMPLE-MATCH 'm8-port-liq-outlet 'm8-tempout-after 'port-liq-d-6
'outlet-temperature-2771)
OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned		notes	Vector-list of matching Equipment Objects Returned (See NOTE) vector #1	vector #2
0	m8- tempout-after	opt-P	outlet-temperature-2771	outlet-temperature-2771
1	m8- port- liq -outlet	p-obj	t770	t770
2	m&edge-outlet	p-obj	t769	t769
3	m8- tempout- before	req-p	outlet-temperature-2766	outlet-temperature-2766
4	m&liquid	p-obj	t758	t758
5	m&port-nozzle -liquid	p-obj	port- onoz-1	port-onoz-4*
6	m8-port-liq-1-inlet	P-W	t771	t771
7	m8- port- liq-2-inlet	p-obj	t775	t775
8	m8- port- liq -impeller	p-obj	t759	t759
9	m&outlet -nozzle	p-obj	r15-outlet-nozzle	r15-outlet-nozzle
10	m8-coldflow-after	W-P	cold-inlet-flow -2771	cold- inlet-flow -2771
11	m8-coldflow- before	W-P	cold-inlet-flow-2766	cold- inlet-flow -2766
12	m8-hotflow -after	req-p	hot -inlet -flow -2771	hot -inlet -flow-2771
13	m8-hotflow -before	W-P	hot -inlet -flow -2766	hot -inlet -flow -2766
14	m8- stream -liq -impeller	p-obj	t762	t762
15	m&hot-cold-mixer	p-obj	t291	t291
16	m8- port- nozzle -floor	p-obj	port- onoz-2	port- onoz-2
17	m8- port-impeller-liquid	p-obj	t761	t761
18	m&tank-floor	p-obj	r15-tank-floor	r15-tank-floor
19	m8- impeller	p-obj	r15- impeller	r15- impeller
20	m&stream-floor-outlet	p-obj	stream -tfloor-onoz	stream -tfloor-onoz
21	m8- port-floor-outnoz	p-obj	port-tfloor-4	port-tfloor-4
22	m8- outflow -after	opt-P	nil	nil
23	m&outflow -before	opt-p	nil	nil
24	m8- height- before	opt-p	nil	nil
25	m8- height-deriv- before	opt-p	nil	nil
26	m&height-after	opt-p	nil	nil
27	m8-cflow-deriv-after	opt-p	nil	nil
28	m8-cflow-deriv-before	opt-p	nil	nil
29	m8-ctemp-deriv- before	opt-p	nil	nil
30	m8-ctemp-deriv-after	opt-p	nil	nil
31	m8-coldtemp -after	opt-p	nil	nil
32	m8-coldtemp -before	opt-p	nil	nil
33	m8- hflowderiv-before	opt-P	nil	nil
34	m8- htemp deriv-after	opt-p	nil	nil
35	m8- hflowderiv-after	opt-p	nil	nil
36	m8- htemp deriv -before	opt-p	nil	nil
37	m8- hottemp-after	opt-p	nil	nil
38	m8- hottemp- before	opt-P	nil	nil
39	m8- neg -port- liquid	neg-1	nil	nil

continued →

* different than vector #1 . p-obj -- positive object neg-n -- member of the nth
req-p -- required parameter opt-p -- optional parameter negative set matched

OUTPUTS: -CONTINUED

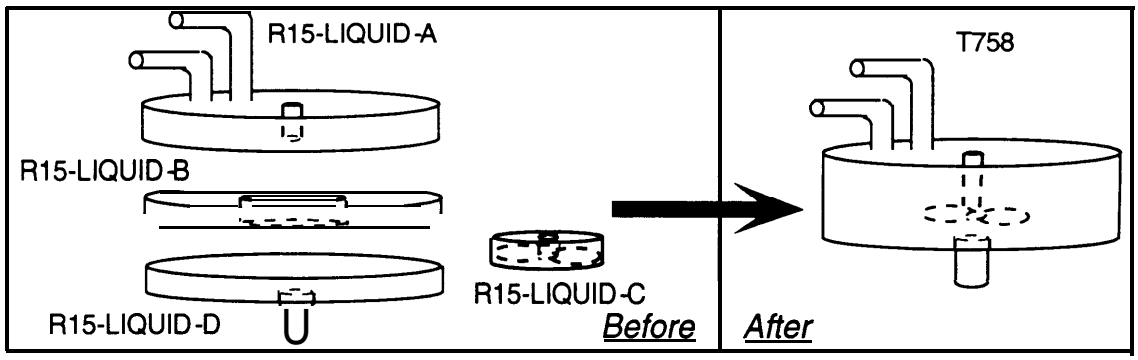
Index from Assoc-list	Vector-list of matching Equipment Objects Returned	
	vector #9	vector #10
0	outlet- temperature -2771	outlet-temperature-2771
1	t770	t770
2	t769	t769
3	outlet-temperature-2766	outlet- temperature -2766
4	t758	t758
5	port- onoz-1	port-onoz-4*
6	t775*	t775*
7	t771.	t771*
8	t759	t759
9	r15-outlet-nozzle	r15-outlet-nozzle
10	hot -inlet -flow -2771 .	hot -inlet -flow -2771*
11	hot -inlet-flow-2766*	hot -inlet -flow -2766*
12	cold- inlet-flow -2771 .	cold- inlet-flow -2771 .
13	cold-inlet-flow -2766*	cold-inlet-flow-2766*
14	t762	t762
15	t291	t291
16	port- onoz-2	port- onoz-2
17	t761	t761
18	r15-tank-floor	r15-tank-floor
19	r15- impeller	r15- impeller
20	stream -tfloor-onoz	stream -tfloor-onoz
21	port-tfloor-4	port-tfloor-4
22	nil	nil
23	nil	nil
24	nil	nil
25	nil	nil
26	nil	nil
27	nil	nil
28	nil	nil
29	nil	nil
30	nil	nil
31	nil	nil
32	nil	nil
33	nil	nil
34	nil	nil
35	nil	nil
36	nil	nil
37	nil	nil
38	nil	nil
39	nil	nil

* different than vector #1 .

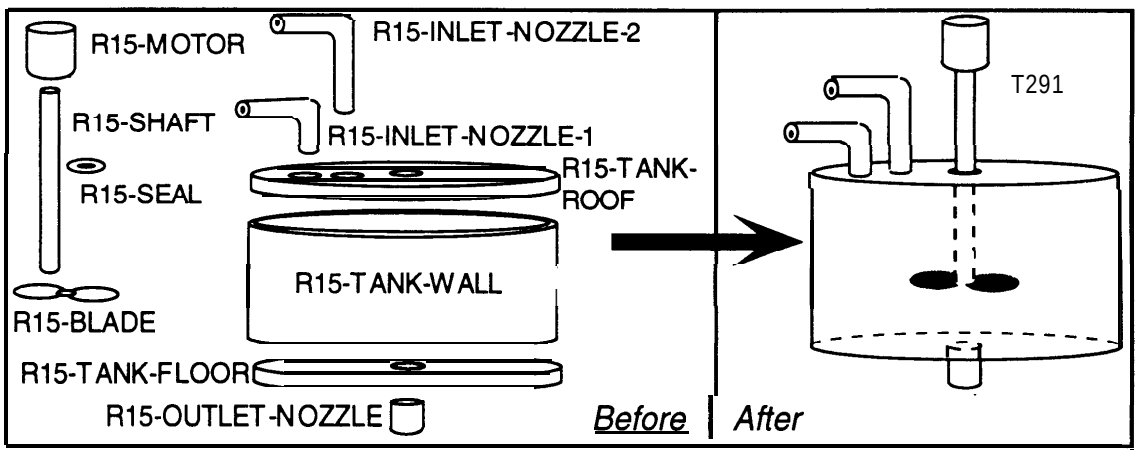
NOTE: 16 vectors (correct matches) were returned, but only the 1 st, 2nd, 9th, and 10th are shown here because all the others do not meet the approximation conditions. See the note attached to the CSTR-1 Matching Case for a description of the 16 vectors.

NOTE: 16 vectors (correct matches) were returned, but only the lst, 2nd, 9th, and 10th are shown here because all the others do not meet the approximation conditions. See the note attached to the CSTR-1 Matching Case for a description of the 16 vectors.

SIDE EFFECTS:



SIDE EFFECTS:--continued



D.10.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 0 vector-list))

OUTPUTS:
T

((m8-dalle-molle::flow-proportional-to-height?
m8- outflow -before value m8- outflow -after value
m&outflow -after value-precision-type m8- height- before value
m&height-after value m&height-after value -precision-type)

(equal m8-hottemp-before value m8- hottemp-after value)

(equal m8-coldtemp-before value m8-coldtemp-after value)

(equal m8-htemp-deriv-after value nil 0)

(equal m 8 htemp -deriv- before value nil 0)

(equal m8-ctemp-deriv-after value nil 0)

(equal m8-ctemp deriv-before value nil 0)

(equal m8-hflow-deriv-after value nil 0)

(equal m8-hflow-deriv-before value nil 0)

(equal m8-height-deriv-before value nil 0)

(equal m8-cflow-deriv-after value nil 0)

'**(equal** m8-cflow-deriv-before value nil 0)

(**m8- dalle-molle:: qspaces-have-common -element ?**
m8- coldtemp -before value -precision-type nil 2
m8- tempout- before value -precision-type nil 2)

(**m8-dalle-molle:: qspaces-have-common-element?**
m8- hottemp- before value-precision-type nil 2
m8- tempout- before value -precision-type nil 4)

(**m8-dalle-molle:: qspace-ok?** m&height-before value-precision-type nil 3)

(possibly-equal?
m8- coldflow-after time m8-coldflow-after time-type
m8-coldflow-after time-interval-type mkoldflow-after time-precision
m8-coldflow-after time-precision-type m&outflow-after time
m&outflow -after time-type m&outflow -after time -interval-type
m&outflow -after time -precision m8- outflow -after time-precision-type)

(**possibly-equal?**
m8-coldflow-after time m8-ccldflow-after time-type
mkoldflow-after time-interval-type m8- coldflow-after time-precision
m8-coldflow-after time -precision-type m8- htemp deriv -after time
m8- htemp deriv-after time -type m8- htemp deriv-after time -interval-type
m8- htemp deriv -after time -precision m8- htempderiv-after time-precision-type)

OUTPUTS:-continued

(possibly -equal?

m8-coldflow-after	time	mkoldflow-after	time -type
m8-coldflow-after	time-interval-type	m8-coldflow-after	time -precision
m8-coldflow-after	time-precision-type	m8-ctemp-deriv-after	time
m8- ctemp-deriv-after	time -type	m8-ctemp-deriv-after	time-interval-type
m8-ctemp-deriv-after	time-precision	m8-ctemp-deriv-after	time-precision-type)

(possibly-equal?

m8-coldflow-after	time	mkoldflow-after	time -type
m8-coldflow-after	time-interval-type	m8-coldflow-after	time -precision
m8- coldflow-after	time -precision-type	m8- hflow -deriv-after	time
m8- hflow deriv -after	time -type	m8- hflow clerk-after	time -interval-type
m8- hflow -deriv-after	time -precision	m8- hflow deriv-after	time -precision-type)

(possiblyequal?

m8- coldflow-after	time	m8-coldflow-after	time-type
m8-coldflow-after	time -interval-type	m8-coldflow-after	time -precision
m8-coldflow-after	time -precision-type	m8- cflow deriv -after	time
m8- cflow -deriv-after	time -type	m8-cflow -deriv-after	time-interval-type
m8-cflow -deriv-after	time -precision	m8-cflow -deriv-after	time-precision-type)

(possibly-equal?

m8- coldflow- before	time	m8- coldflow- before	time -type
m8-coldflow-before	time-interval-type	m8-coldflow- before	time -precision
m8-coldflow- before	time -precision-type	m&outflow -before	time
m&outflow -before	time -type	m8- outflow -before	time -interval-type
m&outflow -before	time -precision	m&outflow -before	time -precision-type)

(possibly-equal?

m8- coldflow- before	time	m8-coldflow- before	time-type
m8- coldf low- before	time -interval-type	m8-coldflow- before	time-precision
m8-coldflow- before	time -precision-type	m8- htemp clerk-before	time
m8- htemp deriv -before	time -type	m8- htemp deriv -before	time -interval-type
m8- htempderiv-before	time -precision	m8- htemp deriv -before	time -precision-type)

(possibly equal?

m8-coldflow- before	time	m8-coldflow- before	time -type
m8-coldflow- before	time -interval-type	m8- coldflow- before	time -precision
m8-coldflow- before	time-precision-type	mktemp-deriv-before	time
m8- ctemp-deriv- before	time -type	m8- ctemp-deriv- before	time -interval-type
m8-ctemp-deriv-before	time-precision	m8-ctemp-deriv- before	time -precision-type)

(possibly-equal?

m8-coldflow- before	time	m8-coldflow- before	time-type
m8-coldflow- before	time -interval-type	m8-coldflow- before	time -precision
m8-coldflow- before	time -precision-type	m8- hflow deriv -before	time
m8- hflow deriv -before	time-type	m8- hflowderiv-before	time-interval-type
m8- hflow deriv -before	time -precision	m8- hflow deriv -before	time -precision-type)

OUTPUTS:—continued

(**possibly-equal?**
m8- coldflow- before time m8-coldflow- before time-type
m8-coldflow- before time -interval-type m8-coldflow- before time -precision
m8-coldflow- before time -precision-type m8- height-deriv- before time
m8- height-deriv- before time -type m8-height-deriv- before time-interval-type
m8- height-deriv-before time-precision m&height-de+ before time-precision-type)

(possibly-equal?
m8-coldflow- before time m8-coldflow- before time -type
m8-coldflow- before time-interval-type m8- coldflow- before time -precision
m8-coldflow- before time -precision-type m8- cflow deriv-before time
m8- cflow deriv -before time -type m8-cflow-deriv-before time-interval-type
m8- cflow deriv-before time -precision m8-cflow-deriv-before time-precision-type)

(possibly-equal?
m8-coldflow- before time m8-coldflow- before time-type
m8-coldflow- before time-interval-type m8-coldflow- before time -precision
m8-coldflow- before time -precision-type m8-height-before time
m8- height- before time -type m8- height- before time -interval-type
m8- height- before time -precision m8- height- before time -precision-type)

(possibly-before?
m8-coldtemp -before value m8- coldtemp -before value -type
m8-coldtemp-before value -interval-type m8- coldtemp -before value -precision
m8-coldtemp -before value -precision-type m8- hottemp- before value
m8- hottemp- before value -type m8- hottemp- before value-interval-type
m8- hottemp- before value -precision m8- hottemp- before value -precision-type)

(equal m&height-after time-units **m8-coldflow-** before time -units)

(equal m8- htemp **-deriv-after** time -units **m8-coldflow** -before time -units)

(equal m8-htemp-deriv-before time-units **m8-coldflow-before** time-units)

(equal m8-ctemp-deriv-after time-units **m8-coldflow-before** time-units)

(equal m8-ctemp-deriv- before time-units **m8-coldflow-** before time-units)

(equal m8- hflow deriv-after time -units **m8-coldflow-before** time-units)

(equal m8- hflowderiv-before time-units **m8-coldflow-before** time-units)

(equal m8- height-deriv- before time -units **m8-coldflow-** before time-units)

(equal m8-cflow deriv -after time -units **m8-coldflow** -before time -units)

(equal m8-cflow-deriv-before time-units **m8-coldflow-before** time-units))

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 1 vector-list)) ← 2nd vector
T
[The list of unverified conditions returned is the same as for the first vector.]

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 8 vector-list)) ← 9th vector
T
[The list of unverified conditions returned is the same as for the first vector.]

(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 9 vector-list)) ← 10th vector
T
[The list of unverified conditions returned is the same as for the first vector.]

D.10.5 Executing the Model

This model does not get executed because the requirement that the M8-height-before model parameter (an input, required parameter) be matched was suspended, and thus no value for the height is available.

D.10.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	**notes**	**Reconfiguration type**	**Reconfigurations tried**
0 m8-tempout-after	opt-P		
1 m8-port-liq outlet	p-obj		
2 m&edge-outlet	p-obj		
3 m8-tempout-before	req-p		
4 m&liquid	p-obj		
5 m8-port-nozzle-liquid	p-obj		
6 m&port-liq-l-inlet	p-obj		
7 m8-port-liq-2-inlet	p-obj		
8 m8-port-liq-impeller	p-obj		
9 m&outlet-nozzle	p-obj		
		intensive	(r15-liquid-d r15-liquid-c) (r15-liquid-d p183-liq) (r15-liquid-d r15-liquid-b) (r15-liquid-d r15-liquid-c r15-liquid-b) (r15-liquid-d r15-liquid-c r15-liquid-a)
1 0 m8-coldflow-after	req-P		
11 m8-coldflow-before	W-P		
12 m8-hotflow-after	W-P		
13 m8-hotflow-before	req-p		
14 m&stream-liq-impeller	p-obj		
		part/whole	(r15-tank-floor r15-outlet-nozzle r15-tank-wall r15-tank-top r15-inlet-nozzle-l r15-inlet-nozzle-2 r15-impeller r15-motor)

req-P -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

Assoc-list of model objects (indicates order of matching)	**notes**	**Reconfiguration type**	**Reconfigurations tried**
1 5 m8-hot-cold-mixer	p-obj		
16 m&port-nozzle -floor	p-obj		
17 m&port-impeller-liquid	p-obj		
18 m8- tank-floor	p-obj		
19 m8- impeller	p-obj		
20 m&stream -floor-outlet	p-obj		
21 m&port-floor-outnoz	p-obj		
22 m&outflow -after	opt-p		
23 m&outflow -before	opt-p		
24 m8- height- before	opt-p		
25 m8- height-deriv- before	opt-p		
26 m8- height-after	opt-p		
2 7 m8-cflow-deriv-after	opt-p		
2 8 m8-cflow-deriv-before	opt-p		
29 m8- ctemp-deriv- before	opt-p		
3 0 m8-ctemp-deriv-after	opt-p		
3 1 m8-coldtemp-after	opt-p		
32 m8-coldtemp -before	opt-p		
33 m8- hflow deriv -before	opt-p		
34 m8- htemp deriv -after	opt-p		
35 m8- hflow-deriv-after	opt-p		
36 m8- htempderiv-before	opt-p		
37 m8- hottemp-after	opt-p		
38 m8- hottemp- before	opt-p	intensive	(r 15-liquid-d r15- liquid-c p183-liq) (r 15-liquid-d p 183-liq r15-liquid-b) (r 15-liquid-d r15- liquid-b r15- liquid-a] (r 15-liquid-d r15- liquid-c r 15-liquid-a p181-liq) (r15-liquid-d r15- liquid-c r 15-liquid-a p182-liq) (r 15-liquid-d r15- liquid-b r15- liquid-a r15-liquid-c)
39 m8- neg -port-liquid	neg-1		

req-p -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

D.11 Dalle-Molle Model and CSTR-3 Equipment Description

D.ll.l Statistics on the Matching

Equipment Description: CSTR-3
Model Description: DALLE-MOLLE
Goal: calculate VALUE

Max. partial matches: 2048
Complete Matches: 16
Rematches: 2
Run time: 2:04 (hr:min)
Objects created (kept): ? (192)
Links created (kept): ? (965)
Separately triggered reconfigurations: 2

Used simple matching? Yes
Used negative matching? Yes
Used intensive reconfiguration? No
Used part-whole reconfiguration? Yes

Positive object triggered reconfiguration? Yes
Parameter triggered reconfiguration? No
Negative set triggered reconfiguration? No

D.11.2 Generating the Potential Match Set

INPUTS:
(SELECT-MODELS 'outlet-temperature-2771 'value 'describes 'port-liq-1 'values)

OUTPUTS;
(
(outlet -temperature-2771 value describes port-liq -1 values m8-tempout -after m8- port-liq -outlet)
)

D.11.3 Match/Reconfigure

INPUTS:
(DO-SIMPLE-MATCH 'm&port-liq-outlet 'm8-tempout-after 'port-liq-1 'outlet-temperature-2771)

OUTPUTS: [association- list and vector- list]

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned (See NOTE) vector #1	vector #5
0 m8- tempout-after	req-p	outlet-temperature-2771	outlet-temperature -2771
1 m8- port- liq -outlet	p-obj	t367	t367
2 m8- edge- outlet	p-obj	edge- onoz-1	edge-onoz-1
3 m8- tempout- before	req-p	outlet-temperature-2766	outlet-temperature -2766
4 m&liquid	p - o b j	t346	t346
5 m8- port-nozzle -liquid	p-obj	port- onoz-1	port- onoz-1
6 m8- port-liq -1 -inlet	p - o b j	t358	t347*
7 m8- port-liq -2-inlet	p-obj	t347	t358*
8 m8- port- liq -impeller	p-obj	t337	t337
g m8- height- before	req-p	liquid- height-2766	liquid- height-2766
10 m8-outlet-nozzle	p-obj	r14-outlet-nozzle	r14- outlet- nozzle
11 m8-coldflow-after	req-p	cold-inlet-flow -2771	hot -inlet -flow -2771*
12 m8-coldflow- before	req-p	cold- inlet-flow -2766	hot -inlet -flow -2766*
13 m8-hotflow -after	req-p	hot -inlet -flow -2771	cold- inlet-flow -2771*
14 m8-hotflow -before	req-p	hot -inlet -flow -2766	cold- inlet-flow -2766*
15 m8- stream -liq -impeller	p-obj	t338	t338
1 6 m8-hot-cold-mixer	p-obj	t233	t233
17 m8- port- nozzle -floor	p-obj	port- onoz-2	port- onoz-2
18 m8- port-impeller-liquid	p-obj	t335	t335
19 m&tank -floor	p-obj	r14-tank-floor	r14-tank-floor
20 m8- impeller	p-obj	t334	t334
21 m&stream-floor-outlet	p-obj	stream -tfloor-onoz	stream -tfloor-onoz
22 m8- port-floor-outnoz	p-obj	port-tfloor-4	port-tfloor-4
23 m8- outflow -after	opt-p	nil	nil
24 m&outflow -before	opt-p	nil	nil
25 m8- height-deriv- before	opt-p	nil	nil
26 m8- height-after	opt-p	nil	nil
27 m8- cflow deriv -after	opt-p	nil	nil
28 m8- cflow deriv -before	opt-p	nil	nil
29 m8-ctemp-deriv- before	opt-p	nil	nil
30 m8-ctemp-deriv-after	opt-p	nil	nil
31 m8-coldtemp -after	opt-p	nil	nil
32 m8-coldtemp -before	opt-p	nil	nil
33 m8- hflow deriv -before	opt-p	nil	nil
34 m8- htempderiv-after	opt-p	nil	nil
35 m8- hflow deriv-after	opt-p	nil	nil
36 m8- htempderiv-before	opt-p	nil	nil
37 m8- hottemp-after	opt-p	nil	nil
38 m8- hottemp- before	opt-p	nil	nil
39 m8- neg -port- liquid	neg-1	nil	nil

• different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-p -- required parameter opt-p -- optional parameter negative set matched

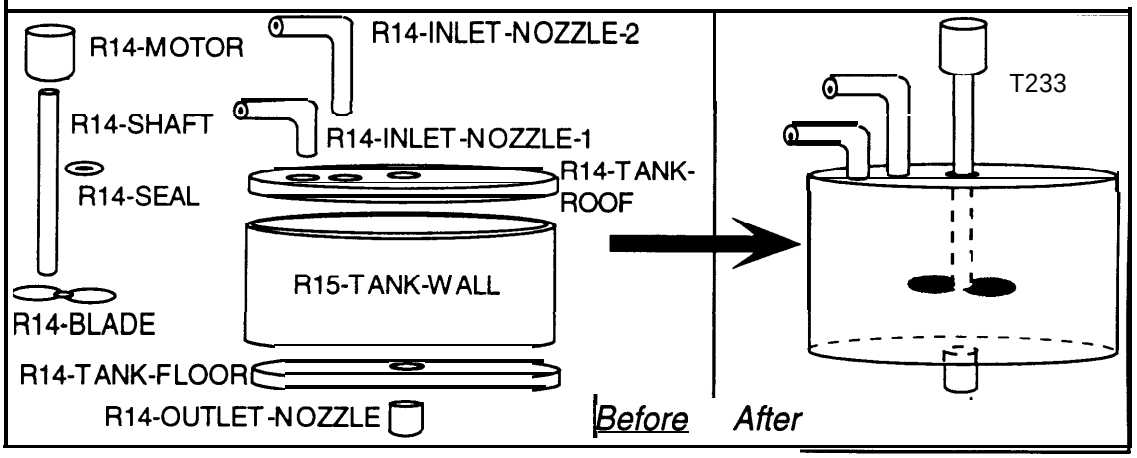
OUTPUTS: -CONTINUED

Index from Assoc-list	Vector-list of matching Equipment Objects Returned	
	vector #9	vector #13
0	outlet- temperature -2771	outlet-temperature -2771
1	t367	t367
2	edge- onoz-1	edge- onoz-1
3	outlet-temperature-2766	outlet- temperature -2766
4	t346	t346
5	port-onoz-4*	port-onoz-4*
6	t358	t347*
7	t347	t358*
8	t337	t337
9	liquid- height-2766	liquid- height-2766
10	r14- outlet- nozzle	r14-outlet-nozzle
11	cold- inlet-flow -2771	hot -inlet -flow-2771 .
12	cold-inlet-flow-2766	hot -inlet -flow -2766*
13	hot -inlet -flow -2771	cold-inlet-flow -2771 .
14	hot -inlet -flow -2766	cold- inlet-flow -2766*
15	t338	t338
16	t233	t233
17	port-onoz-2	port- onoz-2
18	t335	t335
19	r14-tank-floor	r14- tank-floor
20	t334	t334
21	stream -tfloor-onoz	stream -tfloor-onoz
22	port-tfloor-4	port-tfloor-4
23	nil	nil
24	nil	nil
25	nil	nil
26	nil	nil
27	nil	nil
28	nil	nil
29	nil	nil
30	nil	nil
31	nil	nil
32	nil	nil
33	nil	nil
34	nil	nil
35	nil	nil
36	nil	nil
37	nil	nil
38	nil	nil
39	nil	nil

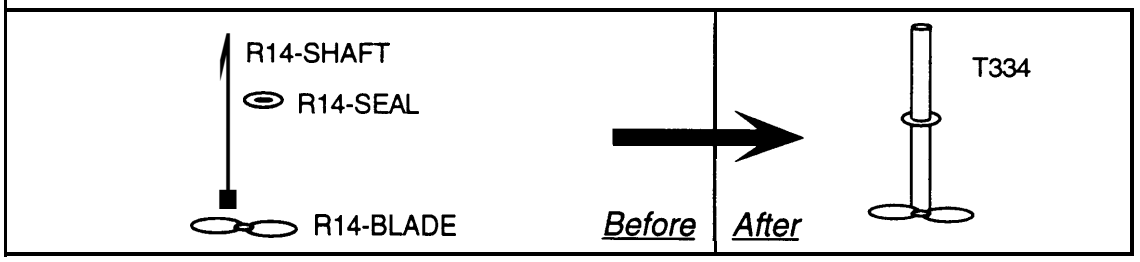
* different than vector #1.

NOTE: 16 vectors (correct matches) were returned, but only the lst, 5th, 9th, and 13th are shown here because all the others do not meet the approximation conditions. See the note attached to the CSTR-1 Matching Case for a description of the 16 vectors.

SIDE EFFECTS:



SIDE EFFECTS:--continued



D.11.4 Checking Model Conditions

INPUTS: Vector #1
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS: Vector #1
T
[The list of unverified conditions returned is identical to the list returned in the CSTR-1 matching.]

INPUTS: Vector 5
(CHECK-MODEL-#CONDITIONS 'approx-conditions assoc-list (fifth vector-list))

OUTPUTS: Vector #5
T
[The list of unverified conditions returned is identical to the list returned in the CSTR-1 matching.]

INPUTS: Vector #9
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (ninth vector-list))

OUTPUTS: Vector #9
T
[The list of unverified conditions returned is identical to the list returned in the CSTR-1 matching.]

INPUTS: Vector #13
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (nth 12 vector-list))

OUTPUTS: Vector #13
T
[The list of unverified conditions returned is identical to the list returned in the CSTR-1 matching.]

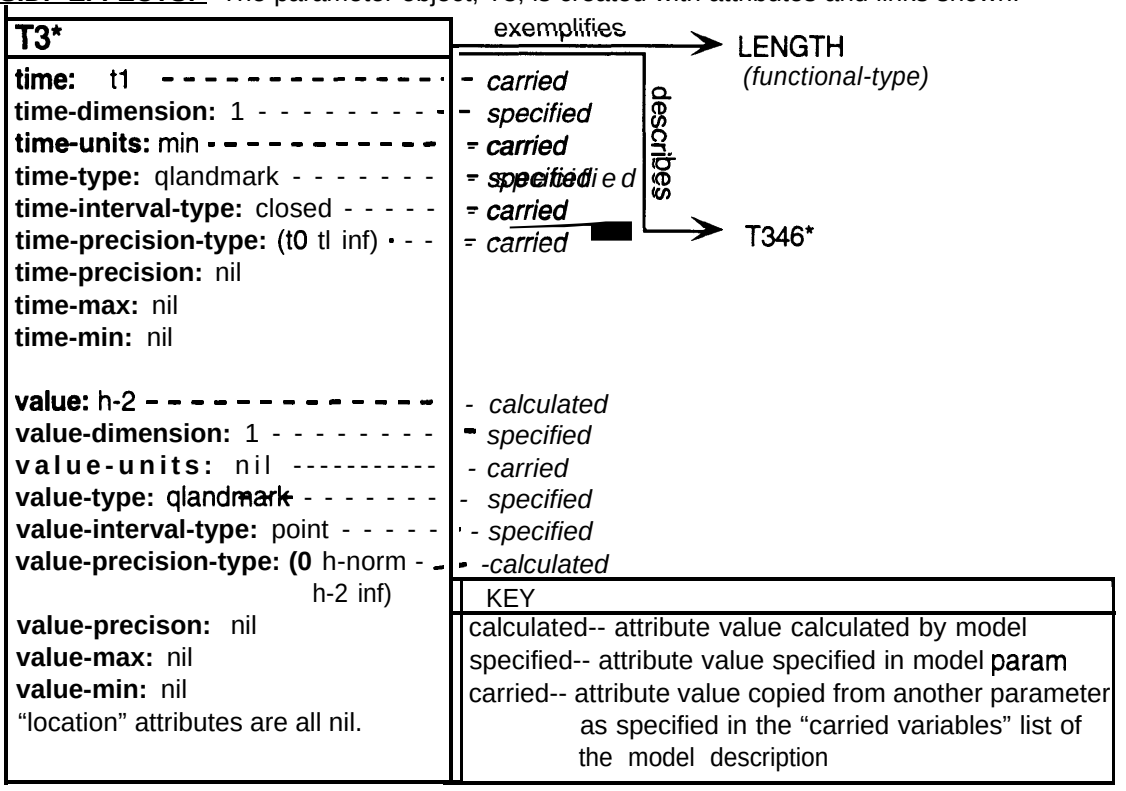
D.11.5 Executing the Model

INPUTS:
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS: [note: The list of QSIM files loaded is printed out to the screen is identical to the list shown in the CSTR-1 matching case.]

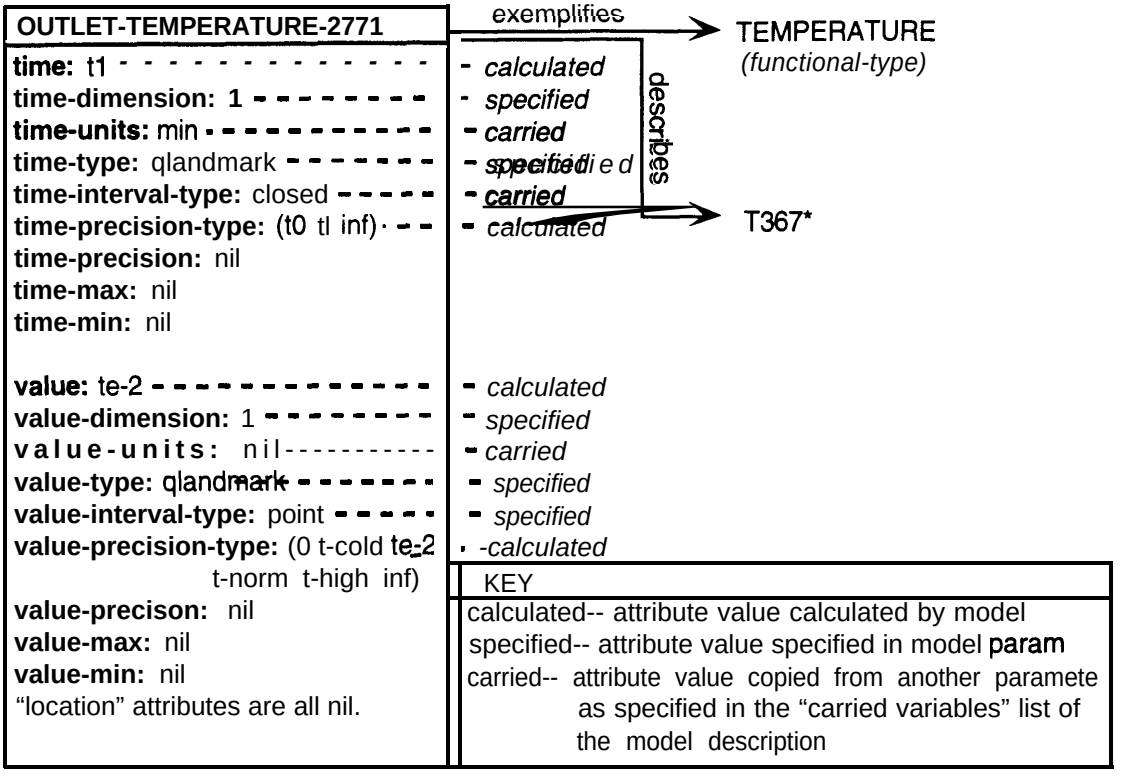
(OUTLET-TEMPERATURE-2771 T3)

SIDF-EFFECTS: The parameter object, T3, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

SIDE-EFFECTS: (continued) The parameter object, outlet-temperature-2771, is modified with attributes and links shown.



- name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.ll.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfigur-at ion type	Reconfigurations tried
0 m8-tempout-after	req-p		
1 m8-port-liq-outlet	p-obj		
2 m&edge-outlet	p-obj		
3 m8-tempout-before	req-p		
4 m&liquid	p-obj		
5 m8-port-nozzle-liquid	p-obj		
6 m&port-liq-l-inlet	p-obj		
7 m8-port-liq-2-inlet	p-obj		
8 m8-port-liq-impeller	p-obj		
9 m&height-before	req-p		
10 m&outlet-nozzle	p-obj		
11 m8-coldflow-after	req-p		
12 m8-coldflow-before	req-p		
13 m8-hotflow-after	req-p		
14 m8-hotflow-before	req-p		
15 m&stream-liq-impeller	p-obj	part/whole	(r14-tank-floor r14-outlet-nozzle r14-tank-wall r14-tank-top r14-inlet-nozzle-1 r14-inlet-nozzle-2 r14-seal r14-shaft r14-impeller-blade r14-motor)
16 m&hot-cold-mixer	p-obj		
17 m8-port-nozzle-floor	p-obj		
18 m&port-impeller-liquid	p-obj		
19 m&tank-floor	p-obj	part/whole	(r14-seal r14-shaft r14-impeller-blade
20 m&impeller	p-obj		
21 m8-stream-floor-outlet	p-obj		
22 m8-port-floor-outnoz	p-obj		
23 m&outflow-after	opt-p		
...	...		
39 m8-neg-port-liquid	neg-1		

req-P -- required parameter
p-obj -- positive object
opt-p -- optional parameter
neg-n -- member of the nth negative set matched

D.12 NPSH Model and DETAILED-PUMP Equipment Description

D.12.1 Statistics on the Matching

Equipment Description:	DETAILED-PUMP
Model Description:	NPSH
Goal:	find EFFECTS
Max. partial matches:	24
Complete Matches:	2
Rematches:	1
Run time:	0:08 (hr:min)
Objects created (kept):	244 (28)
Links created (kept):	1594 (164)
Separately triggered reconfigurations:	
Used simple matching?	*Yes*
Used negative matching?	*Yes*
Used intensive reconfiguration?	*No*
Used part-whole reconfiguration?	*Yes*
Positive object triggered reconfiguration?	*Yes*
Parameter triggered reconfiguration?	*No*
Negative set triggered reconfiguration?	*No*

D.12.2 Generating the Potential Match Set

INPUTS;
(SELECT-MODELS 'water-temperature 'value 'describes 'p76-liquid 'effects)

OUTPUTS;
(water-temperature value describes p76-liquid effects m3-water-temperature m3- internal -water)

D.12.3 Match/Reconfigure

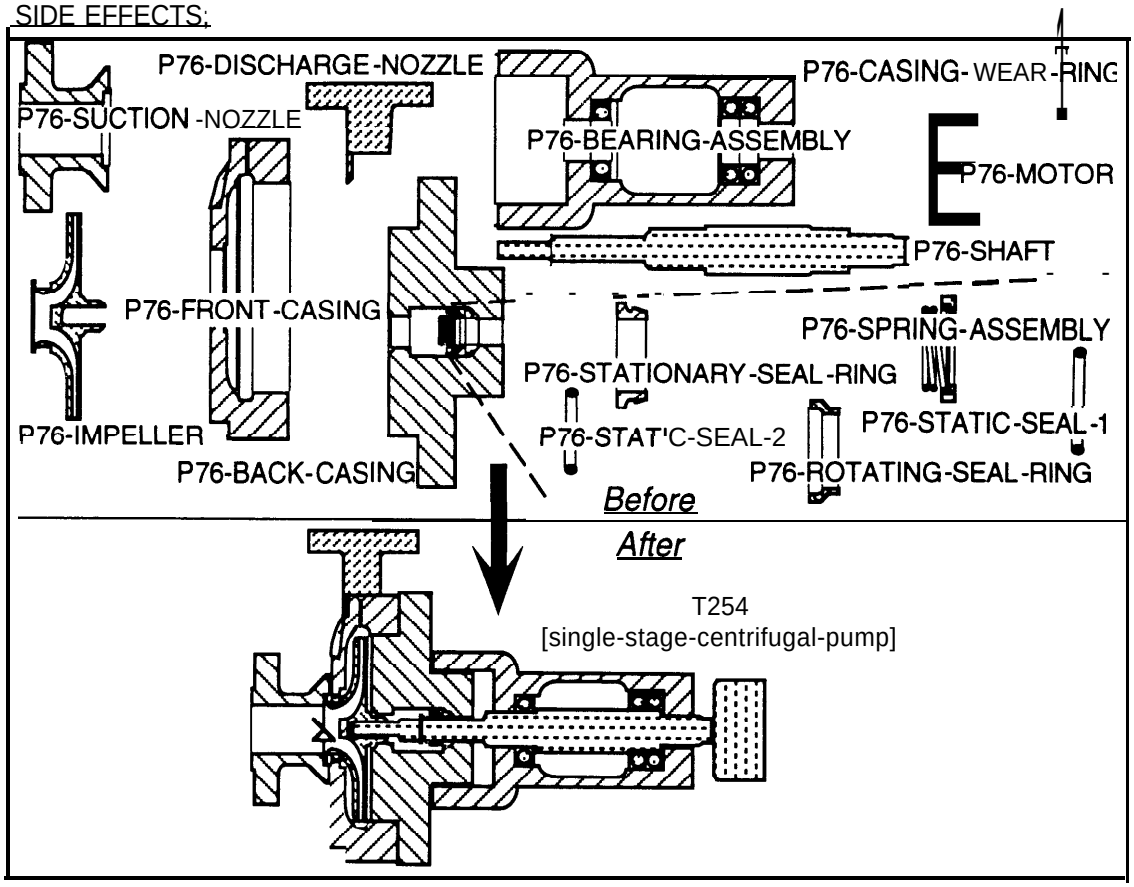
INPUTS:
(DO-SIMPLE-MATCH 'm3-internal-water 'm3-water-temperature 'p76-liquid 'water-temperature)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned		notes	Vector-list of matching Equipment Objects Returned	
			vector #1	vector #2
0	m3 water -temperature	W-P	water-temperature	water-temperature
1	m3 internal -water	p-obj	t267	t267
2	m3 port-l b-inlet	p-obj	t268	t270
3	m3 port-3 a-outlet	p-obj	t270	t268
4	m3 port-2 a	p-obj	t258	t258
5	m3- stream -1 -inlet	p-obj	stream -liquid-l 82liq	stream -liquid-l 83liq
6	m3 inlet -pressure	req-p	pump-inlet -pressure	pump-outlet -pressure
7	m3 inlet -flow -rate	W-P	water-flow	pump-output -flow
8	m3 edge-l b	p-obj	t269	t271
9	m3 stream -3-outlet	p-obj	stream -liquid-l 83liq	stream-liquid-l 82liq
10	m3-edge-3a	p-obj	t271	t269
11	m3 stream -2	p-obj	t259	t259
12	m3-port-2b	p-obj	t255	t255
13	m3-centrifugal-pump	p-obj	t254	t254
14	m3 s-s-impeller	p-obj	p76-impeller	p76-impeller
15	m3 pump-speed	req-p	pump-rpm	pump-rpm
16	m3 water density	opt-p	nil	nil
17	m3-npsh	opt-p	nil	nil
18	m3 inlet -surface-normal	opt-p	nil	nil
19	m3 outlet -surface -normal	opt-p	nil	nil
20	m3 int -water-neg -port	neg -1	nil	nil
21	m3-pump-neg-port-a	neg -2	nil	nil
22	m3 pump-neg-stream-a	neg -2	nil	nil
23	m3 adjacent -water -a	neg -2	nil	nil
24	m3-adj -water-a-neg-port-l	neg -2	nil	nil
25	m3 adj -water-a- neg-port-2	neg -2	nil	nil
26	m3 adj -water-a-neg-stream	neg -2	nil	nil
27	m3-neg-edge-1b	neg 3	nil	nil
28	m3-pump-neg-port-b	neg -4	nil	nil
29	m3-pump-neg-stream-b	neg -4	nil	nil
30	m3 adjacent -water-b	neg -4	nil	nil
31	m3 adj -water-b neg- port-l	neg -4	nil	nil
32	m3- adj -water-b neg- port-2	neg -4	nil	nil
33	m3 adj -water-b neg-stream	neg -4	nil	nil
34	m3-neg-edge-3a	neg -5	nil	nil
35	m3-neg-edge-2a	neg -6	nil	nil
36	m3-neg-endpt-1b	neg -7	nil	nil
37	m3-neg-endpt-3a	neg -8	nil	nil

• different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matched

SIDE EFFECTS:



T254
[single-stage-centrifugal-pump]

D.12.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS approx-conditions assoc-list (first vector-list))

OUTPUTS:
T

((any-common-portion?

m3- inlet -flow -rate	time	m3 inlet -flow-rate	time -type
m3 inlet -flow -rate	time -interval-type	m3 inlet -flow -rate	time -precision
m3 inlet -flow -rate	time -precision-type	m3- water density	time
m3 water density	time -type	m3 water density	time -interval-type
m3- water density	time -precision	m3 water density	time -precision-type)

(any-common-portion?

m3- inlet -flow -rate	time	m3 inlet -flow -rate	time -type
m3 inlet -flow -rate	time -interval-type	m3- inlet -flow -rate	time -precision
m3 inlet -flow -rate	time -precision-type	m3- npsh	time
m3- npsh	time-type	m3- npsh	time -interval-type
m3 npsh	time -precision	m3 npsh	time -precision-type)

(m3- hw-pump ::vector-valuesdont -overlap?

m3 outlet -surface -normal	value	m3 outlet -surface -normal	value -type
m3- outlet surface -normal	interval-type	m3- outlet -surface -normal	value dimension
m3 outlet -surface -normal	value -precision	m3 outlet -surface -normal	value -precision-type
nil	(0 0 0)	nil	real
nil	point	nil	**3**
nil	0	nil	absolute)

(m3- hw-pump ::vector-valuesdont -overlap?

m3 inlet -surface -normal	value	m3 inlet -surface -normal	value -type
m3- inlet -surface-normal	interval-type	m3- inlet -surface -normal	value dimension
m3 inlet -surface-normal	value -precision	m3 inlet -surface-normal	value -precision-type
nil	(0 0 0)	nil	real
nil	point	nil	**3**
nil	0	nil	absolute)

(m3-hw-pump:: time-type-real-or-integer? m3-npsh time-type)

(m3-hw-pump:: time-type-real-or-integer? m3-water-density time-type)

(equal m3 inlet -flow -rate time-units m3 npsh time -units)

(equal m3 inlet -flow -rate time -units m3 water density time- units))

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (second vector-list))

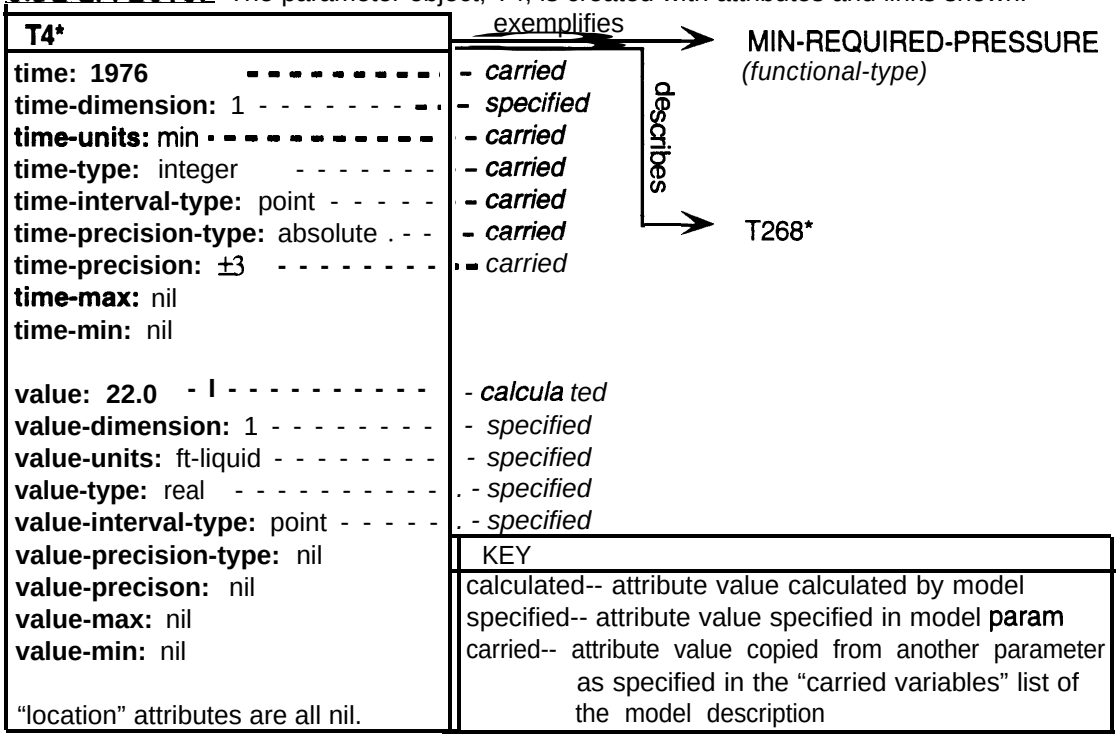
OUTPUTS;
nil

D.12.5 Executing the Model

INPUTS:
(EXECUTE-MODEL assoc-list (first vector-list))

OUTPUTS:
(T4)

SIDE-EFFECTS: The parameter object, T4, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.12.6 Sunimary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfig type	Reconf igurations tried
0 m3 water -temperature	req-P		
1 m3 internal -water	p-obj		
2 m3-port-1 b-inlet	p-obj		
3 m3- port-3a-outlet	p-obj		
4 m3 port-2 a	p-obj		
5 m3 stream -1 -inlet	p-obj		
6 m3- inlet -pressure	req-P		
7 m3 inlet -flow -rate	req-P		
8 m3-edge-1 b	p-obj		
9 m3 stream -3-outlet	p-obj		
10 m3-edge-3a	p-obj		
11 m3 stream -2	p-obj	part/whole-- note #1	(p76-front-casing p76-casing-wear-ring p76-discharge-nozzle p76-impeller p76-suction -nozzle p76- static-seal -2 p76-back-casing p76-static-seal-1 p76-stationary-seal-ring p76- shaft p76-rotating-seal-ring p76-spring-assembly p76-bearing-assembly)
12 m3-port-2b	p-obj		
13 m3-centrifugal-pump	p-obj		
I 4 m3-s-s-impeller	p-obj		
15 m3 pump-speed	req-P		
16 m3 water density	opt-p		
17 m3-npsh	opt-p		
18 m3 inlet -surface-normal	opt-p		
19 m3 outlet -surface -normal	opt-p		
20 m3- int -water -neg -port	neg -1		
21 m3-pump-neg-port-a	neg -2		
22 m3-pump-neg-stream-a	neg -2		
23 m3 adjacent -water -a	neg -2		
24 m3 adj -water- a- neg- port-l	neg -2		
25 m3- adj -water- a- neg- port-2	neg -2		
26 m3- adj -water-a- neg-stream	neg -2		
27 m3-neg-edge-1 b	neg-3		
28 m3-pump-neg-port-b	neg-4		
29 m3-pump-neg-stream-b	neg-4		
30 m3 adjacent -water-b	neg-4		
31 m3 adj -water-b neg- port-l	neg 4		
32 m3 adj -water-b neg- port-2	neg 4		
33 m3 adj -water-b neg-stream	neg 4		
34 m3 neg -edge-3a	neg-5		
35 m3 neg edge-2 a	neg 6		
36 m3- neg -endpt-1b	neg -7		
37 m3- neg -endpt-3a	neg-8		

req-p -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

Note # 1: This part/whole reconfiguration is not triggered by inability to find an equipment object to match M3-PORT-2B, but by the attmpt to find a "has-edge" link from objects maching M3-PORT-2A (4) to objects matching M3-EDGE-1B (8). At this point the matcher was finding all objects linked to M3-PORT-2A. It successfully found and matched M3-STREAM-2, but could not correctly match the "has-edge" link to the objects matching M3-EDGE-1B, which was already found as edge of the port M3-PORT-3A-OUTLET. The surface of the liquid region adjacent to the pump (M3-PORT-2B) must share edges with the ends of the outlet port (M3-PORT-3A-OUTLET), and this condition is not met. Some surface to a pump part has been matched to M3-PORT-2B, and to get the correct surface, the part/whole reconfiguration has to be made.

D.13 Wear Ring Model and DETAILED-PUMP Equipment Description

D.13.1 Statistics on the Matching

Equipment Description: DETAILED-PUMP
Model Description: WEAR-RING
Goal: find CAUSES

Max. partial matches: 13
Complete Matches: 2
Rematches: 1
Run time: 0:10 (hr:min)
Objects created (kept): 261 (50)
Links created (kept): 1546 (275)
Separately triggered reconfigurations: 1

Used simple matching? Yes
Used negative matching? Yes
Used intensive reconfiguration? No
Used part-whole reconfiguration? Yes

Positive object triggered reconfiguration? Yes
Parameter triggered reconfiguration? No
Negative set triggered reconfiguration? No

D.13.2 Generating the Potential Match Set

INPUTS;
(SELECT-MODELS 'power-input-72 'value 'describes 'p76-motor 'causes)

OUTPUTS:
(
(power-input -72 value describes p76-motor causes m5- power-2 m5- motor)
)

D.13.3 Match/Reconfigure

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
0 m5-power-2	req-p	power-input-72	power-input-72
1 m5-motor	p-obj	p76-motor	p76-motor
2 m5-port-motor-pump	p-obj	port-motor-pump	port-motor-pump
3 m5-power-l	req-p	power-input-l	power-input-l
4 m5-stream-pump-motor	p-obj	stream-motor-pump	stream-motor-pump
5 m5-port-pump-motor	p-obj	t231	t231
6 m5-c-pump	p-obj	t217	t217
7 m5-port-pump-liq	p-obj	t223	t223
8 m5-imp-wear-ring	p-obj	p76-imp-wear-ring	p76-imp-wear-ring
9 m5-cas-wear-ring	p-obj	p76-casing-wear-ring	p76-casing-wear-ring
10 m5-impeller	p-obj	p76-impeller	p76-impeller
11 m5-stream-pump-liq	p-obj	t226	t226
12 m5-edge-fluid-inlet	p-obj	t219	t224
13 m5-edge-fluid-outlet	p-obj	t224	t219
14 m5-i-wear-ring-od	req-p	impeller-wear-ring-od	impeller-wear-ring-od
15 , m5-c-wear-ring-id	req-p	casing-wear-ring-id	casing-wear-ring-id
16 m5-port-liq-pump	p-obj	t225	t225
17 m5-port-fluid-inlet	p-obj	t235	t237
18 m5-port-fluid-outlet	p-obj	t237	t235
19 m5-fluid	p-obj	t234	t234
20 m5-clearance	opt-P	nil	nil
21 m5-rpm-1	opt-P	pump-rpm	pump-rpm
22 m5-rpm-2	opt-P	nil	nil
23 m5-rho-2	opt-P	nil	nil
24 m5-rho-1	opt-p	nil	nil

* different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matchec

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
25 m5-mu-2	opt-p	nil	nil
26 m5-mu-1	opt-p	nil	nil
27 m5-pump-neg-port-a	neg -1	nil	nil
28 m5- pump- neg-stream-a	neg -1	nil	nil
29 m5- adjacent -fluid -a	neg -1	nil	nil
30 m5- adj -a- neg- port-l	neg -1	nil	nil
31 m5- adj -a- neg- port-2	neg -1	nil	nil
32 m5- adj -a- neg-stream	neg-1	nil	nil
33 m5- pump- neg-port-b	neg -2	nil	nil
34 m5- pump-neg-stream-b	neg -2	nil	nil
35 m5-adjacent-fluid-b	neg -2	nil	nil
36 m5- adj -b- neg- port-l	neg -2	nil	nil
37 m5- adj -b- neg- port-2	neg -2	nil	nil
38 m5- adj -b- neg-stream	neg -2	nil	nil
39 m5- neg edge-fluid-inlet	neg-3	nil	nil
40 m5- neg -edge-fluid-outlet	neg-4	nil	nil
41 m5- neg -port-fluid	neg-5	nil	nil

different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-p -- required parameter opt-p -- optional parameter negative set matchec

D.13.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS approx-conditions assoc-list (first vector-list))

OUTPUTS:

T

((any-common -portion ?

m5- power-2	time	m5- power-2	time-type
m5- power-2	time -interval-type	m5- power-2	time -precision
m5- power-2	time -precision-type	m5- rpm-2	time
m5- rpm-2	time-type	m5- rpm-2	time -interval-type
m5- rpm-2	time -precision	m5- rpm-2	time-precision-type)

(any-common-portion?

m5- power-2	time	m5- power-2	time-type
m5- power-2	time-interval-type	m5- power-2	time -precision
m5- power-2	time-precision-type	m5-rho-2	time
m5-rho-2	time-type	m5-rho-2	time-interval-type
m5-rho-2	time -precision	m5-rho-2	time -precision-type)

(any-common-portion?

m5- power-2	time	m5- power-2	time-type
m5- power-2	time -interval-type	m5- power-2	time -precision
m5- power-2	time -precision-type	m5-mu-2	time
m5-mu-2	time-type	m5-mu-2	time -interval-type
m5-mu-2	time -precision	m5-mu-2	time -precision-type)

(any-common-portion?

m5- power-l	time	m5- power-l	time -type
m5- power-l	time-interval-type	m5- power-l	time -precision
m5- power-l	time -precision-type	m5-rho-1	time
m5-rho-1	time-type	m5-rho-1	time -interval-type
m5-rho-1	time -precision	m5-rho-1	time -precision-type)

, (any -common -portion?

m5- power-l	time	m5- power-l	time-type
m5- power-l	time -interval-type	m5- power-l	time -precision
m5- power-l	time -precision-type	m5-mu-1	time
m5-mu-1	time-type	m5-mu-1	time -interval-type
m5-mu-1	time -precision	m5-mu-1	time -precision-type)

(any -common -portion?

m5-rpm-1	value	m5-rpm-1	value -type
m5-rpm-1	value-interval-type	m5 rpm-1	value -precision
m5-rpm-1	value -precision-type	m5- rpm-2	value
m5- rpm-2	value -type	m5- rpm-2	value -interval-type
m5- t-pm-2	value -precision	m5 rpm-2	value -precision-type)

(any-common-portion?

m5-rho-1	value	m5-rho-1	value -type
m5-rho-1	value -interval-type	m5-rho-1	value -precision
m5-rho-1	value -precision-type	m5-rho-2	value
m5-rho-2	value -type	m5-rho-2	value -interval-type
m5-rho-2	value -precision	m5-rho-2	value-precision-type)

(any-common-portion?

m5-mu-1	value	m5-mu-1	value -type
m5-mu-1	value -interval-type	m5-mu-1	value -precision
m5-mu-1	value-precision-type	m5-mu-2	value
m5-mu-2	value -type	m5-mu-2	value -interval-type
m5-mu-2	value -precision	m5-mu-2	value -precision-type)

(equal m5-rpm-1 value-units m5-rpm-2 value-units)

(equal m5-mu-1 value-units m5-mu-2 value-units)

(equal m5-rho-1 value-units m5-rho-2 value-units))

INPUTS:

(CHECK-MODEL-CONDITIONS approx-conditions assoc-list (second vector-list))

OUTPUTS:

T

((any-common-portion ?

m5-power-2	time	m5-power-2	time-type
m5-power-2	time -interval-type	m5-power-2	time -precision
m5-power-2	time -precision-type	m5-rpm-2	time
m5-rpm-2	time -type	m5-rpm-2	time -interval-type
m5-rpm-2	time -precision	m5-rpm-2	time -precision-type)

(any-common-portion?

m5-power-2	time	m5-power-2	time -type
m5-power-2	time -interval-type	m5-power-2	time -precision
m5-power-2	time -precision-type	m5-rho-2	time
m5-rho-2	time-type	m5-rho-2	time -interval-type
m5-rho-2	time -precision	m5-rho-2	time -precision-type)

(any-common -portion?

m5-power-2	time	m5-power-2	time -type
m5-power-2	time -interval-type	m5-power-2	time -precision
m5-power-2	time -precision-type	m5-mu-2	time
m5-mu-2	time -type	m5-mu-2	time -interval-type
m5-mu-2	time -precision	m5-mu-2	time -precision-type)

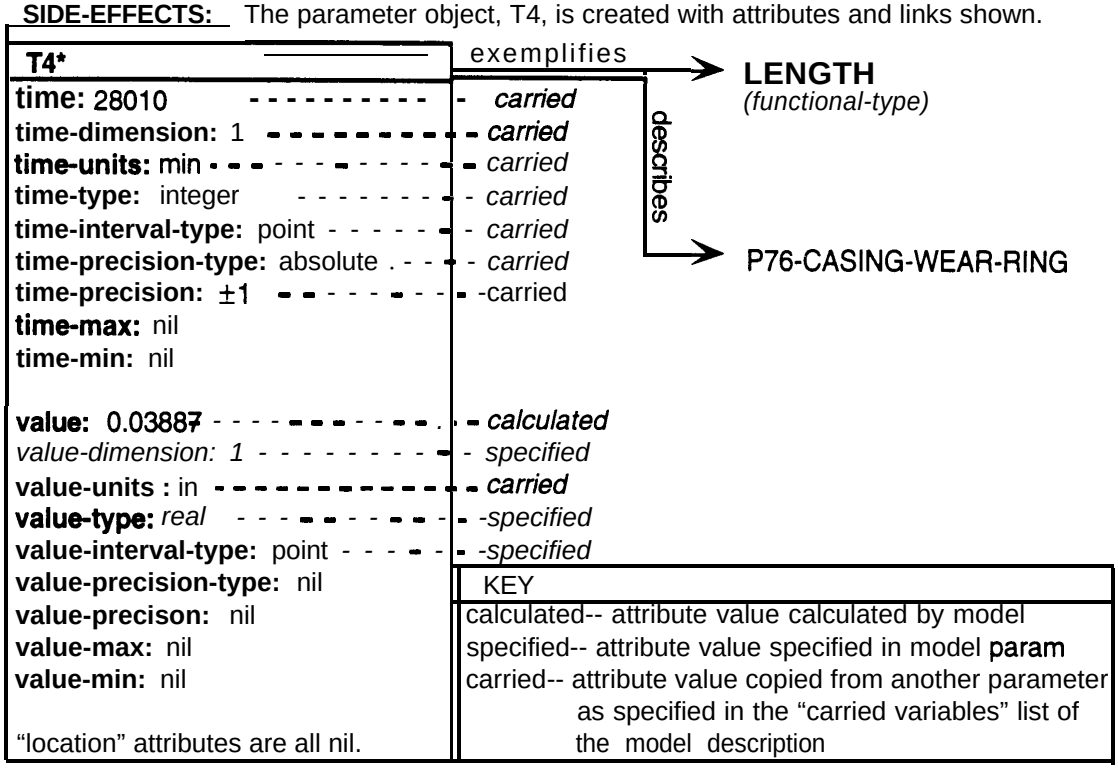
```
(any-common-portion?
 m5- power-l      time
 m5- power-l      time -interval-type
 m5- power-l      time -precision-type
 m5-rho-1         time -type
 m5-rho-1         time -precision
 m5- power-l      time -type
 m5- power-l      time -precision
 m5-rho-1         time
 m5-rho-1         time -interval-type
 m5-rho-1         time -precision-type)

(any-common-portion?
 m5- power-l      time
 m5- power-l      time -interval-type
 m5- power-l      time -precision-type
 m5-mu-1          time-type
 m5-mu-1          time -precision
 m5- power-l      time -type
 m5- power-l      time -precision
 m5-mu-1          time
 m5-mu-1          time -interval-type
 m5-mu-1          time -precision-type)

(any-common -portion?
 m5-rpm-1         value
 m5-rpm-1         value -interval-type
 m5-rpm-1         value -precision-type
 m5-rpm-2         value -type
 m5- rpm-2        value -precision
 m5-rpm-1         value -type
 m5-rpm-1         value -precision
 m5- rpm-2        value
 m5- rpm-2        value -interval-type
 m5- rpm-2        value -precision-type)

(any-common -portion?
 m5-rho-1         value
 m5-rho-1         value -interval-type
 m5-rho-1         value-precision-type
 m5-rho-2         value -type
 m5-rho-2         value-precision
 m5-rho-1         value -type
 m5-rho-1         value -precision
 m5-rho-2         value
 m5rho-2          value -interval-type
 m5-rho-2         value-precision-type)

(any-common-portion?
 m5-mu-1          value
 m5-mu-1          value -interval-type
 m5-mu-1          value-precision-type
 m5- mu -2        value -type
 m5- mu -2        value -precision
 m5-mu-1          value -type
 m5-mu-1          value -precision
 m5-mu-2          value
 m5-mu-2          value -interval-type
 m5-mu-2          value -precision-type)

(equal m5- rpm-1 value -units m5-rpm -2 value-units)

(equal m5- mu -1 value-units m5- mu -2 value -units)

(equal m5-rho-1 value-units m5-rho-2 value-units))
```

D.13.5 Executing the Model

INPUTS:
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS:
(T4)

SIDE-EFFECTS: The parameter object, T4, is created with attributes and links shown.



• name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.13.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
0 m5-power-2	req-p		
1 m5-motor	p-obj		
2 m5-port-motor-pump	p-obj		
3 m5-power-1	req-p		
4 m5- stream -pump-motor	p-obj		
5 m5- port-pump-motor	p-obj	part/whole	(p76-shaft p76-stationary-seal-ring p76-rotating-seal-ring p76-static-seal-2 p76-static-seal-1 p76-spring-assembly p76-impeller p76-bearing-assembly p76-back-casing p76-front-casing p76-casing -wear-ring p76-discharge-nozzle p76-suction nozzle)
6 m5-c-pump	p-obj		
7 m5-port-pump-liq	p-obj		
8 m5- imp-wear- ring	p-obj		
9 m5-cas-wear-ring	p-obj		
10 m5- impeller	p-obj		
11 m5- stream -pump-liq	p-obj		
12 m5- edge-fluid-inlet	p-obj		
1 3 m5-edge-fluid-outlet	p-obj		
1 4 m5-i-wear-ring-od	req-p		
1 5 m5-c-wear-ring-id	req-p		
1 6 m5-port-liq-pump	p-obj		
17 m5- port-fluid-inlet	p-obj		
18 m5- port-fluid-outlet	p-obj		
1 9 m5-fluid	p-obj		
2 0 m5-clearance	opt-P		
2 1 m5-rpm-1	opt-p		
2 2 m5-rpm-2	opt-p		

req-P -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

D.14 Hydraulic Horsepower Model and DETAILED-PUMP Equipment

D.14.1 Statistics on the Matching

Equipment Description:	DETAILED-PUMP
Model Description:	HYDRAULIC-HORSEPOWER
Goal:	find EFFECTS
Max. partial matches:	***59***
Complete Matches:	
Rematches:	1
Run time:	0:11 (hr:min)
Objects created (kept):	273 (31)
Links created (kept):	1851 (194)
Separately triggered reconfigurations:	
Used simple matching?	Yes
Used negative matching?	Yes
Used intensive reconfiguration?	No
Used part-whole reconfiguration?	Yes
Positive object triggered reconfiguration?	Yes
Parameter triggered reconfiguration?	No
Negative set triggered reconfiguration?	No

D.14.2 Generating the Potential Match Set

INPUTS;
(SELECT-MODELS 'pump-inlet-pressure 'value 'describes 'port-liq-13 'effects)

OUTPUTS:
(
(pump-inlet-pressure value describes port-liq-13 effects m&suction-pressure m6-port-liq-inlet)
(pump-inlet-pressure value describes port-liq -13 effects m6-discharge-pressure
m&port-Q-outlet)
)

D.14.3 Match/Reconfigure

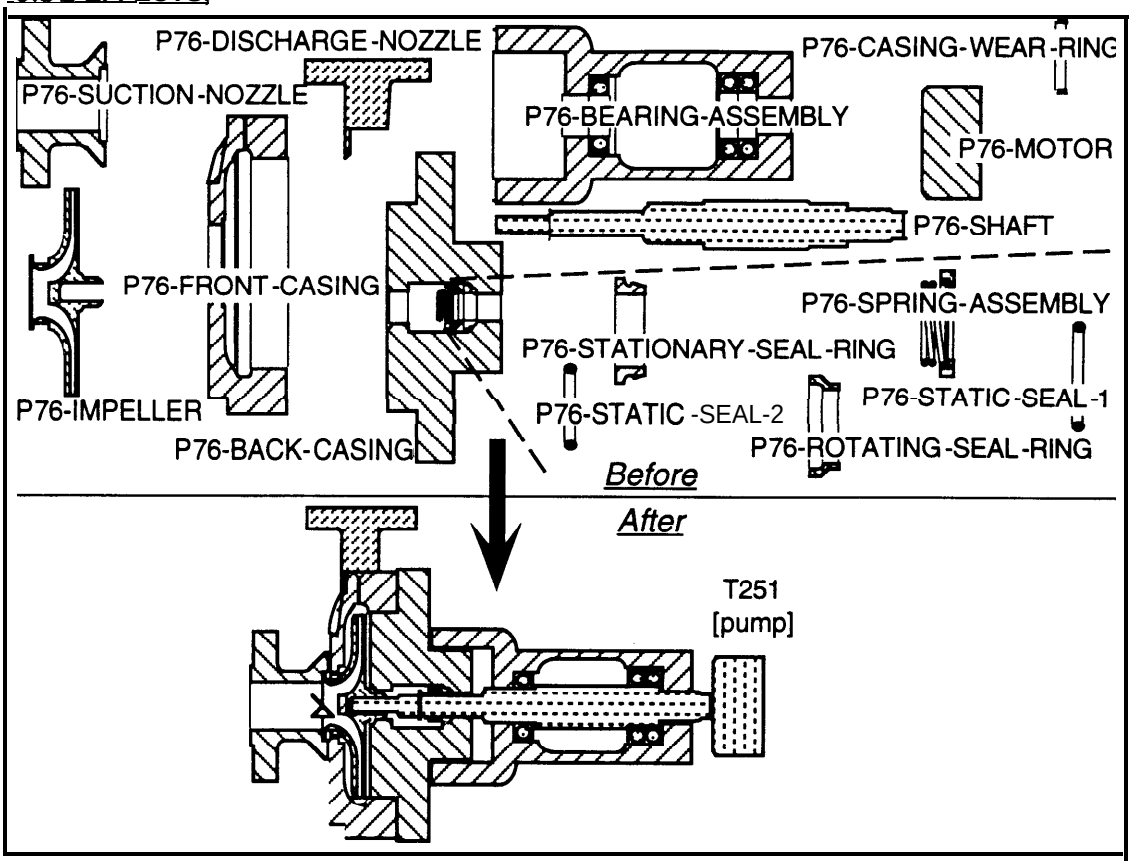
INPUTS:
(DC-SIMPLE-MATCH 'm6-port-liq-inlet 'm6-suction-pressure 'port-liq-13 'pump-inlet-pressure)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
0 m6-suction-pressure	W-P	pump-inlet-pressure	
1 m6-port-liq-inlet	p-obj	t265	
2 m6-liquid	p-obj	t264	
3 m6-edge-liq-inlet	p-obj	t266	
4 m6-flow-rate	req-P	water-flow	
5 m6-suction-nozzle-id	req-P	pump-inlet diameter	
6 m6-port-liq-outlet	p-obj	t267	
7 m6-port-liq-pump	p-obj	t255	
8 m6-specific-gravity	req-P	water-specific-gravity	
9 m6-port-pump-liq	p-obj	t252	
10 m6-edge-liq-outlet	p-obj	t268	
11 m6-discharge-nozzle-id	req-P	pump-outlet diameter	
12 m6-discharge-pressure	req-P	pump-outlet -pressure	
13 m6-stream-pump-liq	p-obj	t256	
14 m6-pump	p-obj	t251	
15 m6-hydraulic-horsepower	opt-p	nil	
16 m6-neg-edge-liq-inlet	neg-1	nil	
17 m6-adj-a-neg-stream	neg-2	nil	
18 m6-pump-neg-port-a	neg-2	nil	
19 m6-pump-neg-stream-a	neg-2	nil	
20 m6-adjacent-liq-a	neg-2	nil	
21 m6-adj-a-neg-port-l	neg-2	nil	
22 m6-adj-a-neg-port-2	neg-2	nil	
23 m6-neg-port-liq	neg-3	nil	
24 m6-neg-edge-liq-outlet	neg-4	nil	
25 m6-adj-b-neg-stream	neg 5	nil	
26 m6-pump-neg-port-b	neg-5	nil	
27 m6-pump-neg-stream-b	neg-5	nil	
28 m6-adjacent-liq-b	neg-5	nil	
29 m6-adj-b-neg-port-l	neg-5	nil	
30 m6-adj-b-neg-port-2	neg-5	nil	

* different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matched

SIDE EFFECTS;



D.14.4 Checking Model Conditions

INPUTS:

(CHECK-MODEL-CONDITIONS approx-conditions assoc-list (first vector-list))

OUTPUTS:

T

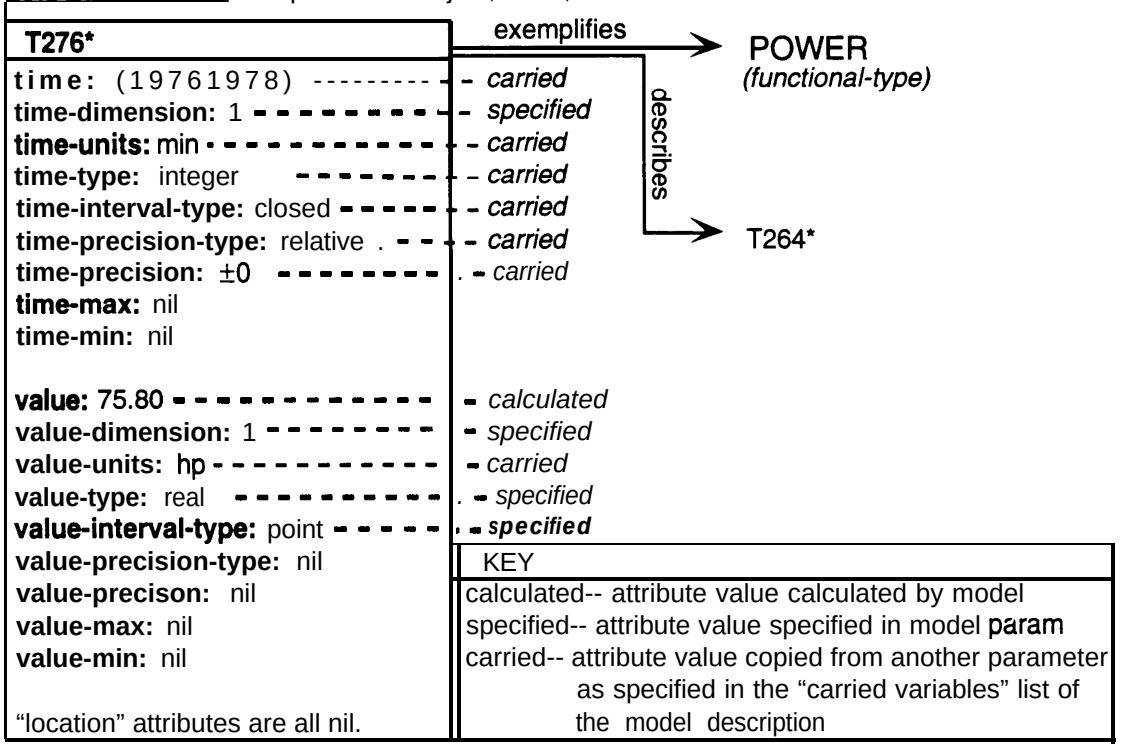
NIL

D.14.5 Executing the Model

INPUTS:
(EXECUTE-MODEL assoc-list (first vector-list))

OUTPUTS:
(T276)

SIDE-EFFECTS: The parameter object, T276, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.14.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
0 m6- suction -pressure	W - P		
1 m6- port- liq -inlet	p-obj		
2 m6- liquid	p-obj		
3 m6- edge- liq -inlet	p-obj		
4 m6- flow -rate	req-p		
5 m6- suction -nozzle -id	req-p		
6 m6- port-liq -outlet	p-obj		
7 m6- port- liq -pump	p-obj		
8 m6- specific -gravity	req-p		
9 m6- port-pump-liq	p-obj		
1 0 m6-edge-liq-outlet	p-obj		
11 m&discharge-nozzle-id	req-p		
12 m6- discharge -pressure	req-p		
13 m6- stream -pump-liq	p-obj	part/whole	(p76-front-casing p76-discharge-nozzle p76-suction -nozzle p76-back-casing p76-casing -wear-ring p76-static-seal-1 p76-stationary-seal-ring p76-rotating-seal-ring p76-static-seal-2 p76-spring-assembly p76-bearing-assembly p76-shaft p76-impeller)
14 m6-pump	p-obj		
15 m6- hydraulic- horsepower	opt-p		
16 m6- neg -edge- liq -inlet	neg -1		
17 m6- adj -a- neg-st ream	neg -2		
18 m6-pump-neg-port-a	neg -2		
19 m6- pump-neg-stream-a	neg -2		
20 m6- adjacent -liq -a	neg -2		
21 m6- adj -a- neg- port-l	neg -2		
22 m6- adj -a- neg- port-2	neg -2		
23 m6- neg -port- liq	neg -3		
24 m6- neg -edge-liq -outlet	neg -4		
25 m6- adj -b- neg- stream	neg -5		
26 m6-pump-neg-port-b	neg -5		
27 m&pump-neg-stream-b	neg -5		
28 m6- adjacent -liq - b	neg -5		
29 m6-adj-b-neg-port-1	neg -5		
30 m6- adj -b- neg- port-2	neg -5		

req-P -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

D.15 Hypothetical Model and DETAILED-PUMP Equipment Description

D.15.1 Statistics on the Matching

Equipment Description:	DETAILED-PUMP
Model Description:	HYPOTHETICAL
Goal:	find EFFECTS
Max. partial matches:	52
Complete Matches:	4
Rematches:	2
Run time:	0:18 (hr:min)
Objects created (kept):	379 (85)
Links created (kept):	2612 (449)
Separately triggered reconfigurations:	2
Used simple matching?	Yes
Used negative matching?	Yes
Used intensive reconfiguration?	No
Used part-whole reconfiguration?	Yes
Positive object triggered reconfiguration?	Yes
Parameter triggered reconfiguration?	No
Negative set triggered reconfiguration?	No

D.15.2 Generating the Potential Match Set

INPUTS:
(SELECT-MODELS 'water-temperature 'value 'describes 'p76-liquid 'effects)

OUTPUTS:
(
(water-temperature value describes p76-liquid effects m3-water-temperature m3-internal -water)
(water-temperature value describes p76-liquid effects m7-fluid -temperature m7-internal -fluid)

D.15.3 Match/Reconfigure

INPUTS:
(DC-SIMPLE-MATCH 'm7-internal-fluid 'm7-fluid-temperature 'p76-liquid 'water-temperature)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
0 m7-fluid -temperature	W-P	water-temperature	water- temperature
1 m7- internal -fluid	p-obj	t262	t262
2 m7- port-2 a	p-obj	t253	t253
3 m7- port-l b-inlet	p-obj	t265	t368*
4 m7-port-3a-outlet	p-obj	t263	t263
5 m7- stream -2	p-obj	t254	t254
6 m7- edge-l b	p-obj	t251	t251
7 m7- edge-3 a	p-obj	t264	t264
8 m7- stream -1 -inlet	p-obj	stream -liquid-l 831iq	stream -liquid-l 831iq
9 m7- inlet -pressure	W-P	pump-outlet -pressure	pump-outlet -pressure
10 m7- inlet -flow -rate	req-P	pump-output -flow	pump-output -flow
11 m7- stream-3-outlet	p-obj	stream -liquid-l 821iq	stream -liquid-l 821iq
12 m7- port-2 b	p-obj	t250	t250
13 m7-cent -pump	p-obj	t249	t249
14 m7- impeller	p-obj	p76-impeller	p76-impeller
15 m7- pump-casing	p-obj	t328	t328
1 6 m7-pump-speed	W-P	pump-rpm	pump-rpm
17 m7-fluid density	opt-p	nil	nil
18 m7- inlet -surface -normal	opt-p	nil	nil
19 m7- npsh	opt-p	nil	nil
20 m7- outlet -surface -normal	opt-P	nil	nil
21 m7- int -water-neg -port	neg -1	nil	nil
2 2 m7-pump-neg-port-a	neg -2	nil	nil
2 3 m7-pump-neg-stream-a	neg -2	nil	nil
24 m7-adj -fluid-a	neg-2	nil	nil

* different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matched

INPUTS:
(DO-SIMPLE-MATCH 'm7-internal-fluid 'm7-fluid -temperature 'p76-liquid 'water-temperature)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #3	vector #4
0 m7-fluid -temperature	W-P	water-temperature	water-temperature
1 m7- internal -fluid	p-obj	t262	t262
2 m7- port-2 a	p-obj	t253	t253
3 m7- port-l b-inlet	p-obj	t263*	t366*
4 m7-port-3a-outlet	p-obj	t265*	t265*
5 m7- stream -2	p-obj	t254	t254
6 m7- edge-l b	p-obj	t252*	t252*
7 m7-edge-3a	p-obj	t266*	t266*
8 m7- stream -1 -inlet	p-obj	stream-liquid-l 82liq .	stream -liquid-l 82liq *
9 m7- inlet -pressure	req-p	pump-inlet -pressure*	pump-inlet -pressure*
10 m7- inlet -flow -rate	req-p	water-flow .	water-flow*
11 m7- stream -3-outlet	p-obj	stream-liquid-l 83liq *	stream-liquid-l 83liq *
12 m7- port-2 b	p-obj	t250	t250
13 m7-cent -pump	p-obj	t249	t249
14 m7- impeller	p-obj	p76-impeller	p76-impeller
15 m7- pump-casing	p-obj	t328	t328
16 m7-pump-speed	req-p	pump-rpm	pump-rpm
17 m7-fluid density	opt-p	nil	nil
18 m7- inlet -surface -normal	opt-p	nil	nil
19 m7- npsh	opt-P	nil	nil
20 m7- outlet -surface -normal	opt-P	nil	nil
21 m7- int -water -neg -port	neg -1	nil	nil
22 m7-pump-neg-port-a	neg -2	nil	nil
23 m7- pump-neg-stream-a	neg -2	nil	nil
24 m7- adj -fluid-a	- neg 2	nil	nil

* different than vector #1. p-obj -- positive object neg-n -- member of the nth negative set matched
req-P -- required parameter opt-p -- optional parameter

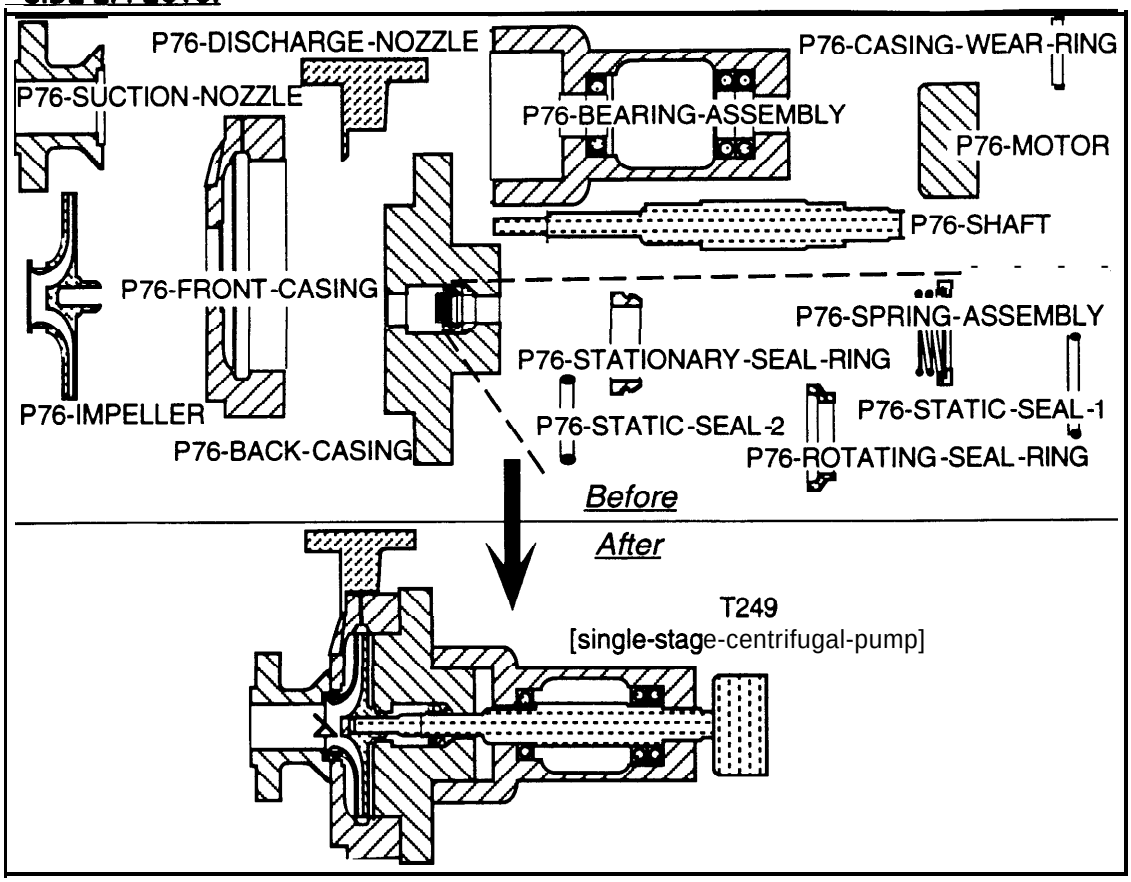
Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
25 m7- adj -a-neg-stream	neg -2	nil	nil
26 m7- adj -a- neg- port-l	neg -2	nil	nil
27 m7- adj -a- neg- port-2	neg -2	nil	nil
2 8 m7-pump-neg-port-b	neg-3	nil	nil
29 m7- pump-neg-stream-b	neg-3	nil	nil
30 m7-adj-fluid- b	neg-3	nil	nil
3 1 m7-adj-b-neg-stream	neg-3	nil	nil
32 m7- adj -b- neg- port-l	neg-3	nil	nil
3 3 m7-adj-b-neg-port-2	neg-3	nil	nil
3 4 m7-neg-edge-2a	neg-4	nil	nil
35 m7-neg-edge-1 b	neg-5	nil	nil
3 6 m7-neg-edge-3a	neg-6	nil	nil
37 m7-neg-endpt-1 b	neg -7	nil	nil
38 m7- neg-endpt-3a	neg-8	nil	nil

* different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matchec

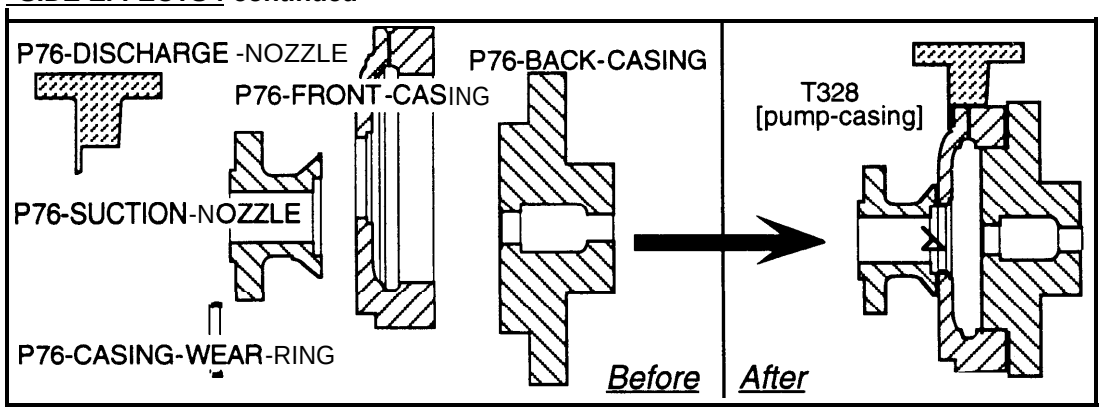
Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #3	vector #4
25 m7- adj -a- neg-stream	neg -2	nil	nil
26 m7- adj -a- neg- port-l	neg -2	nil	nil
27 m7- adj -a- neg- port-2	neg -2	nil	nil
28 m7- pump- neg-port-b	neg-3	nil	nil
29 m7- pump- neg-stream-b	neg-3	nil	nil
30 m7- adj -fluid-b	neg-3	nil	nil
31 m7- adj -b neg-stream	neg-3	nil	nil
32 m7- adj -b neg- port-l	neg-3	nil	nil
33 m7- adj -b neg- port-2	neg-3	nil	nil
34 m7- neg edge-2 a	neg-4	nil	nil
35 m7-neg-edge-1 b	neg-5	nil	nil
36 m7- neg-edge-3 a	neg-6	nil	nil
37 m7- neg-endpt-1 b	neg -7	nil	nil
3 8 m7-neg-endpt-3a	neg-8	nil	nil

• different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matchec

SIDE EFFECTS:



SIDE EFFECTS : continued



D.15.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS:
T

No condition checking because this isn't a real model and thus doesn't have any conditions.

D.15.5 Executing the Model

INPUTS:
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS;
No execution of model because this isn't a real model.

SIDE-EFFECTS: No side effects.

D.15.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	notes	Reconfiguration type	Reconfigurations tried
0 m7-fluid -temperature	req-p		
1 m7- internal -fluid	p-obj		
2 m7- port-2 a	p-obj		
3 m7- port-l b-inlet	p-obj		
4 m7- port-3 a-outlet	p-obj		
5 m7- stream -2	p-obj		
6 m7- edge-l b	p-obj		
7 m7-edge-3a	p-obj		
8 m7- stream-l -inlet	p-obj		
9 m7- inlet -pressure	req-p		
10 m7- inlet -flow -rate	req-p		
11 m7- stream -3-outlet	p-obj		
12 m7-port-2 b	p-obj	part/whole	(p76-static-seal-1 p76-stationary-seal-ring p76-rotating-seal-ring p76-static-seal-2 p76-spring-assembly p76-bearing-assembly p76-shaft p76-impeller p76-casing -wear-ring p76-front-casing p76-back-casing p76-discharge-nozzle p76-suction -nozzle)
13 m7- cent -pump	P-W		
14 m7- impeller	p-obj	part/whole	(p76-casing -wear- ring p76-front-casing p76-back-casing p76-discharge-nozzle p76-suction -nozzle)
15 m7- pump-casing	P-W		
16 m7- pump-speed	req-p		
17 m7-fluid density	opt-p		
18 m7- inlet -surface-normal	opt-p		

req-P -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

D.16 Friction Factor Model and PIPES-l Equipment Description

D.16.1 Statistics on the Matching

Equipment Description:	PIPES-1
Model Description:	FRICTION-FACTOR
Goal:	find EFFECTS
Max. partial matches:	816
Complete Matches:	2
Rematches:	12
Run time:	1:38 (hr:min)
Objects created (kept):	319 (12)
Links created (kept):	1856 (69)
Separately triggered reconfigurations:	2
Used simple matching?	Yes
Used negative matching?	Yes
Used intensive reconfiguration?	Yes
Used part-whole reconfiguration?	No
Positive object triggered reconfiguration?	No
Parameter triggered reconfiguration?	No
Negative set triggered reconfiguration?	Yes

D.16.2 Generating the Potential Match Set

INPUTS;
(SELECT-MODELS 'pipe-2c-roughness 'value 'describes 'pipe-2-port-3c 'effects)

OUTPUTS;
(
(pipe-2c-roughness value describes pipe -2-port-3c effects ml 0- pipe -roughness m10-port-3b)
)

D.16.3 Match/Reconfigure

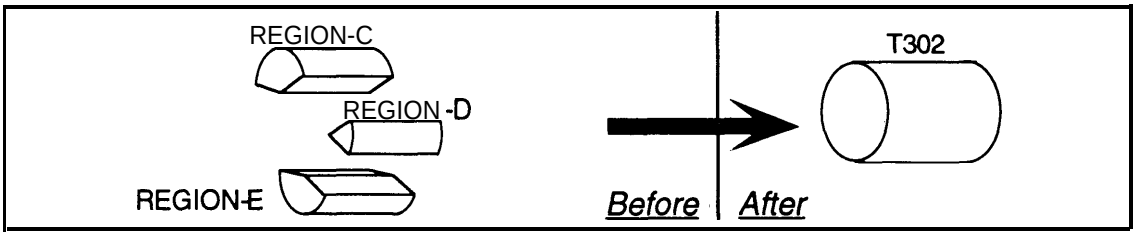
INPUTS:
(DC-SIMPLE-MATCH 'm10-port-3b 'm10-pipe-roughness 'pipe-2-port-3c 'pipe-2c-roughness)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned		notes	Vector-list of matching Equipment Objects Returned	
			vector #1	vector #2
0	ml 0- pipe -roughness	req-p	pipe -2c-roughness	pipe -2 c-roughness
1	ml 0- port3 b	p-obj	t312	t312
2	ml 0- stream3	p-obj	t313	t313
3	m10-pipe	p-obj	pipe -2	pipe -2
4	m10-edge-2a	p-obj	t321	t320*
5	ml 0- edge-l b	p-obj	t320	t321 .
6	ml 0- port3 a	p-obj	t303	t303
7	ml 0- pipe -insidediameter	W-P	pipe-2-id	pipe-2-id
8	m10-port-2a	p-obj	t317	t314*
9	ml 0- port-l b	p-obj	t314	t317*
10	m10-fluid-in-pipe	p-obj	t302	t302
11	ml O-stream-2	p-obj	t319	t316*
12	m10-stream-1	p-obj	t316	t319*
13	ml O-friction-factor	opt-p	nil	nil
14	ml O-outlet-temperature	opt-p	nil	nil
15	ml 0- inlet-temperature	opt-p	nil	nil
16	ml O-velocity	opt-p	2d-velocity	nil .
17	ml 0- density	opt-p	rho-c	□≃□▱◗
18	ml O-viscosity	opt-p	mu-c	○◆▱◗
19	ml 0- neg-endpt-2 a	neg -1	nil	nil
20	m10-neg-endpt-1 b	neg -2	nil	nil
21	m10-neg-edge-3a	neg -3	nil	nil
22	m10-neg-edge-2a	neg -4	nil	nil
23	ml 0- neg-edge-l b	neg -5	nil	nil
24	m10-neg-port	neg -6	nil	nil

. different than vector #1. p-obj -- positive object neg-n -- member of the nth
req-P -- required parameter opt-p -- optional parameter negative set matched

SIDE EFFECTS;



D.16.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS:
T

((possibly-equal?

ml O-outlet- temperature	value	ml O-outlet- temperature	value -type
ml O-outlet- temperature	value -interval-type	ml O-outlet- temperature	value -precision
ml O-outlet-temperature	value-precision-type	ml 0- inlet-temperature	value
ml 0- inlet-temperature	value -type	ml 0- inlet-temperature	value-interval-type
ml 0- inlet-temperature	value -precision	ml 0- inlet-temperature	value -precision-type)

(equal m 1 O-inlet-temperature value-units ml 0- outlet-temperature value -units)

(any-common-portion?

ml O-outlet-temperature	time	ml O-outlet- temperature	time -type
ml O-outlet-temperature	time-interval-type	ml O-outlet- temperature	time -precision
ml O-outlet-temperature	**time-precision-type**	ml O-velocity	time
ml O-velocity	time -type	ml O-velocity	time -interval-type
ml O-velocity	time -precision	ml O-velocity	time -precision-type)

(any-common-portion?

ml 0- inlet-temperature	time	ml 0- inlet-temperature	time-type
ml 0- inlet-temperature	time-interval-type	ml 0- inlet-temperature	time-precision
ml 0- inlet-temperature	time -precision-type	ml O-velocity	time
ml O-velocity	time -type	ml O-velocity	time -interval-type
ml O-velocity	time -precision	ml O-velocity	time -precision-type)

(equal m10-pipe-roughness time-units m10-outlet-temperature time-units)

(equal m10-pipe-roughness time-units ml 0- inlet- temperature time -units))

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (second vector-list))

OUTPUTS:
T

((ml O-friction :: re-and-kd -ok?
ml 0- pipe -inside-diameter value ml O-velocity value
ml O-density value ml O-viscosity value
m10-pipe-roughness value)

(m10-friction::re-ok?
ml O-pipe -insidediameter value ml O-velocity value
ml O-density value ml O-viscosity value)

(possibly-equal?
ml O-outlet-temperature value m10-outlet-temperature value-type
ml O-outlet- temperature value -interval-type ml O-outlet- temperature value -precision
ml 0-outlet-temperature value-precision-type ml 0- inlet-temperature value
ml O-inlet-temperature value -type ml 0- inlet-temperature value -interval-type
ml 0- inlet-temperature value -precision ml 0- inlet-temperature value -precision-type)

(equal m 1 O-inlet-temperature value -units ml O-outlet-temperature value-units)

(any-common -portion?
ml O-outlet-temperature time ml O-outlet-temperature time -type
m10-outlet-temperature time-interval-type ml O-outlet-temperature time -precision
m10-outlet-temperature time-precision-type m10-velocity time
ml O-velocity time-type m10-velocity time -interval-type
ml O-velocity time -precision ml O-velocity time -precision-type)

(any-common-portion?
ml 0- inlet-temperature time ml 0- inlet-temperature time-type
ml 0- inlet-temperature time -interval-type ml 0- inlet-temperature time-precision
ml 0- inlet-temperature time -precision-type ml O-velocity time
ml O-velocity time-type ml O-velocity time -interval-type
ml O-velocity time -precision ml O-velocity time -precision-type)

(any-common-portion?
ml O-viscosity time ml O-viscosity time-type
ml O-viscosity time -interval-type ml O-viscosity time-precision
ml O-viscosity time -precision-type ml O-velocity time
ml O-velocity time-type ml O-velocity time -interval-type
ml O-velocity time -precision ml O-velocity time-precision-type)

(any-common-portion?

ml O-density	time	m10-density	time -type
ml O-density	time -interval-type	ml O-density	time -precision
ml O-density	time-precision-type	ml O-velocity	time
ml O-velocity	time -type	ml O-velocity	time -interval-type
ml O-velocity	time -precision	ml O-velocity	time -precision-type)

(any-common-portion?

ml 0- pipe -inside-diameter	time	ml 0- pipe -inside-diameter	time -type
ml 0- pipe-insidediameter	time -interval-type	ml 0- pipe -inside-diameter	time -precision
ml 0- pipe-inside-diameter	time -precision-type	ml O-velocity	time
ml O-velocity	time -type	ml O-velocity	time -interval-type
ml O-velocity	time -precision	ml O-velocity	time -precision-type)

(any-common -portion?

ml 0- pipe -roughness	time	m10-pipe-roughness	time -type
ml 0- pipe -roughness	time -interval-type	ml 0- pipe-roughness	time -precision
ml O-pipe -roughness	time -precision-type	ml 0-velocity	time
ml O-velocity	time -type	ml O-velocity	time -interval-type
ml O-velocity	time -precision	ml O-velocity	time -precision-type)

(equal ml 0- pipe -roughness time- units ml O-outlet- temperature time-units)

(equal ml O-pipe -roughness time- units ml 0- inlet-temperature time-units))

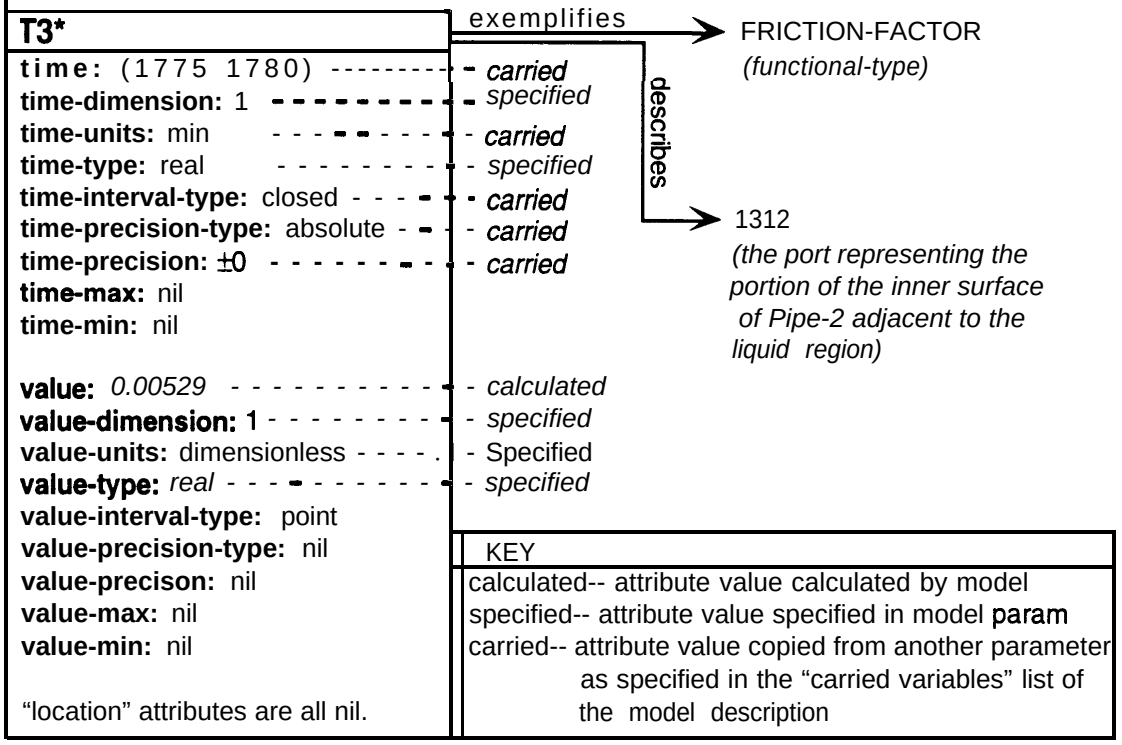
(equal ml 0- pipe-roughness time-units m 1 O-velocity time -units))

D.16.5 Executing the Model

INPUTS:
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS:
(T3)

SIDE-EFFECTS: The Parameter Object, T3, is created with attributes and links shown.



* name of parameter object is created by tha GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.16.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)		notes	Reconfiguration type	Reconfigurations tried
0	m10-pipe-roughness	req-p		
1	m10-port-3b	p-obj		
2	m10-stream-3	p-obj		
3	m10-pipe	p-obj		
4	m10-edge-2a	p-obj		
5	m10-edge-1b	p-obj		
6	m10-port-3a	p-obj		
7	m10-pipe-inside-diameter	req-p		
8	m10-port-2a	p-obj		
9	m10-port-1b	p-obj		
10	m10-fluid-in-pipe	p-obj		
11	m10-stream-2	p-obj		
12	m10-stream-1	p-obj		
13	m10-friction-factor	opt-p		
14	m10-outlet-temperature	opt-p		
15	m10-inlet-temperature	opt-p		
16	m10-velocity	opt-p		
17	m10-density	opt-p		
18	m10-viscosity	opt-p		
			intensive	(region-c region-y)
				(region-c region-b)
19	m10-neg-endpt-2a	neg-1		
			intensive	(region-c region-d)
				(region-c region-e)
				(region-c region-y region-d)
				(region-c region-y region-e)
				(region-c region-y region-b)
				(region-c region-b region-d)
				(region-c region-b region-e)
				(region-c region-b region-a)
				(region-c region-d region-e)
20	m10-neg-endpt-1b	neg-2		
21	m10-neg-edge-3a	neg-3		
22	m10-neg-edge-2a	neg-4		
23	m10-neg-edge-1b	neg-5		
24	m10-neg-port	neg-6		

req-p -- required parameter
opt-p -- optional parameter
p-obj -- positive object
neg-n -- member of the nth negative set matched

D.17 Hagen-Poiseuille Model and PIPES-l Equipment Description

D.17.1 Statistics on the Matching

Equipment Description:	PIPES-1	
Model Description:	HAGEN-POISEUILLE	
Goal:	find EFFECTS	
Max. partial matches:	30	
Complete Matches:	1	
Rematches:	0	
Run time:	0:02	(hr:min)
Objects created (kept):	1 (1)	
Links created (kept):	2 (2)	
Separately triggered reconfigurations:	0	
Used simple matching?	Yes	
Used negative matching?	Yes	
Used intensive reconfiguration?	No	
Used part-whole reconfiguration?	No	
Positive object triggered reconfiguration?	No	
Parameter triggered reconfiguration?	No	
Negative set triggered reconfiguration?	No	

D.17.2 Generating the Potential Match Set

INPUTS:
(SELECT-MODELS 'inlet-pressure-a 'value 'describes 'port-al 'effects)

OUTPUTS:
(
(inlet-pressure-a value describes port-al effects m9-outlet-pressure m9-port-2a)
(inlet-pressure-a value describes port-al effects m9-inlet-pressure m9-port-1 b)
)

D.17.3 Match/Reconfigure

INPUTS:
(DO-SIMPLE-MATCH 'm9-port-1 b 'm9-inlet-pressure 'port-a1 'inlet-pressure-l)

OUTPUTS: (association-list and vector-list)

Assoc-List of Model Objects Returned	notes	Vector-list of matching Equipment Objects Returned	
		vector #1	vector #2
0 m9-inlet-pressure	req-P	inlet-pressure -a	
1 m9-port-1 b	p-obj	port-al	
2 m9-stream -1	p-obj	stream -x2- al	
3 m9-edge-l b	p-obj	pipe -1 -edge-2-1	
4 m9-fluid-in-pipe	p-obj	region -a	
5 m9-port3 b	p-obj	pipe-2-port-3 a	
6 m9-port-3 a	p-obj	port-a3	
7 m9-port-2 a	p-obj	port-a2	
8 m9-viscosity-l	req-p	mu-a	
9 m9-stream -3	p-obj	stream -pipe -2-a3	
1 0 m9-pipe	p-obj	pipe -2	
11 m9-edge-2 a	p-obj	edge-a2-1	
1 2 m9-length	req-P	region -a-length	
1 3 m9-stream-2	p-obj	stream -a2- b 1	
14 m9-outlet -pressure	req-P	outlet- pressure -a	
15 m9-pipe -inside diameter	req-P	pipe-2-id	
16 m9-surface -normal -1 b	opt-p	port- normal-al	
17 m9-vol -flow -rate	opt-P	nil	
18 m9-inlet -pressure -2	opt-p	nil	
19 m9-viscosity -2	opt-p	nil	
2 0 m9-density	opt-p	rho-a	
2 1 m9-shear-stress-1	opt-p	nil	
22 m9-shear-stress-2	opt-p	nil	
23 m9-surface -normal -2 a	opt-P	port- normal-a2	
24 m9-pipe -axial direction	opt-p	pipe-2-axial direction	
25 m9-neg -edge-l b	neg -1	nil	
26 m9-neg -s- normal-l b	neg -2	nil	
27 m9-neg-endpt-1 b	neg-3	nil	
28 m9-neg -port	neg-4	nil	
29 m9-neg-edge-3 a	neg-5	nil	
30 m9-neg edge-2 a	neg-6	nil	
3 1 m9-neg-s-normal-2 a	neg -7	nil	
3 2 m9-neg-endpt-2 a	neg-8	nil	

• different than vector #1 . p-obj -- positive object neg-n -- member of the nth negative set matched
W-P -- required parameter opt-p -- optional parameter

D.17.4 Checking Model Conditions

INPUTS:
(CHECK-MODEL-CONDITIONS 'approx-conditions assoc-list (first vector-list))

OUTPUTS;
T

((m9-hagen :: is-flow-steady-state?
m9- inlet-pressure value m 9- inlet-pressure -2 value)

(m9- hagen ::fluid-is-newtonian?
m9-shear-stress-1 value m9- viscosity -1 value
m9-shear-stress-2 value m9- viscosity -2 value)

(equal m9-shear-stress-1 value -units m9-shear-stress-2 value -units)

(m9- hagen ::flow-is-laminar?
m9- pipe -inside diameter value m9- vol -flow -rate value
m9- density value m9- viscosity -1 value)

(possibly-equal?

m9- pressure deriv	value	m9- pressure deriv	value -type
m9- pressurederiv	value -interval-type	m9- pressure deriv	value -precision
m9- pressurederiv	value -precision-type	nil	0
nil	real	nil	point
nil	nil	nil	nil)

(not-any-common-portion?

m9-shear-stress-1	time	m9-shear-stress-1	time-type
m9- shear-stress-l	time-interval-type	m9-shear-stress-1	time -precision
m9-shear-stress-1	time-precision-type	m9- shear-stress-2	time
m9- shear- stress-2	time-type	m9- shear-stress-2	time-interval-type
m9-shear-stress-2	time -precision	m9-shear-stress-2	time-precision-type)

, (any-common-portion?

m9-shear-stress-2	time	m9-shear-stress-2	time-type
m9-shear-stress-2	time-interval-type	m9- shear-stress-2	time -precision
m9- shear-stress-2	time -precision-type	m9- viscosity -2	time
m9- viscosity -2	time-type	m9- viscosity -2	time-interval-type
m9- viscosity -2	time -precision	m9- viscosity -2	time -precision-type)

(any-common-portion?

m9- inlet -pressure	time	m9- inlet -pressure	time -type
m9- inlet -pressure	time -interval-type	m9- inlet -pressure	time -precision
m9- inlet -pressure	time -precision-type	m9- vol -flow -rate	time
m9- vol -flow -rate	time -type	m9- vol -flow -rate	time -interval-type
m9- vol -flow -rate	time -precision	m9- vol -flow -rate	time -precision-type)

(any- common- portion?

m9-shear-stress-1	time	m9- shear-stress-l	time -type
m9- shear- stress-l	time -interval-type	m9- shear-stress-l	time -precision
m9- shear-stress-l	time -precision-type	m9- viscosity -1	time
m9- viscosity -1	time -type	m9- viscosity -1	time -interval-type
m9- viscosity -1	time -precision	m9- viscosity -1	time -precision-type)

(equal m9-viscosity-1 time -units m9-viscosity-2 time -units)

(equal m9-viscosity-1 time-units m9-shear-stress-2 time-units)

(equal m9-viscosity-1 time-units m9-shear-stress-1 time-units)

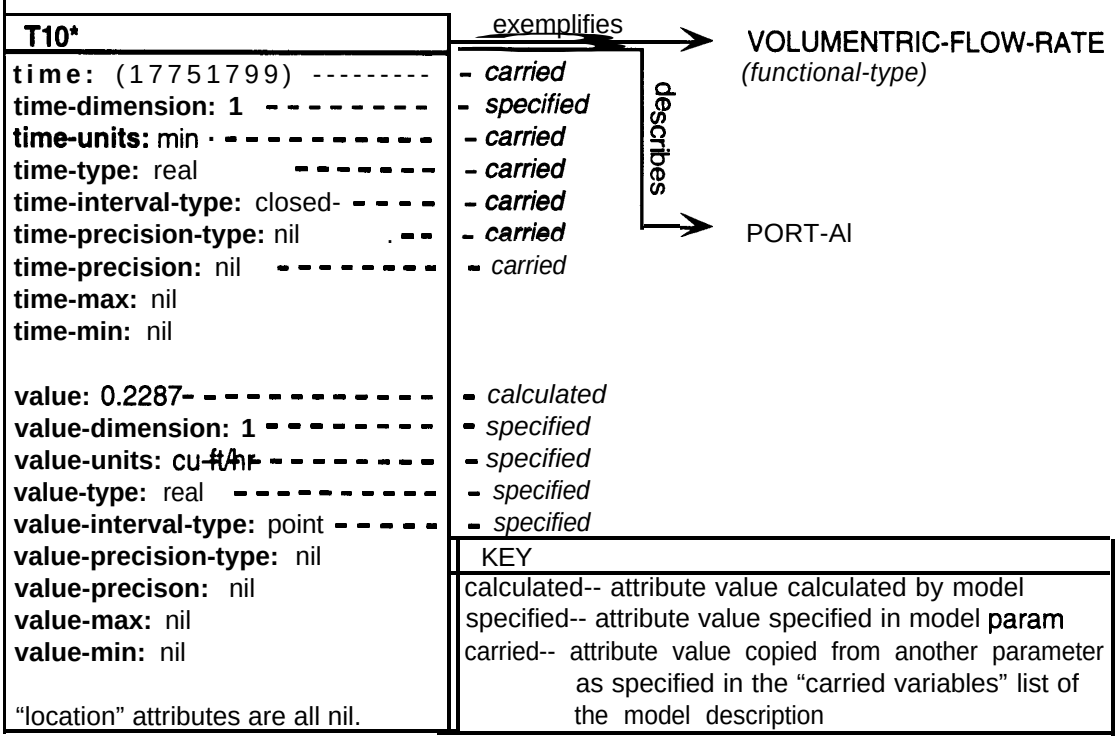
(equal m9-viscosity-1 time-units m9- inlet -pressure -2 time -units))

D.17.5 Executing the Model

INPUTS;
(EXECUTE -MODEL assoc-list (first vector-list))

OUTPUTS;
T10

SIDE-EFFECTS: The parameter object, T10, is created with attributes and links shown.



* name of parameter object is created by the GENTEMP function which generates unique names, all beginning with "T" followed by an integer.

D.17.6 Summary of Merges and Splits Tested

Assoc-list of model objects (indicates order of matching)	**notes**	**Reconfiguration type**	**Reconfigurations tried**
No reconfigurations were made.			

req-P -- required parameter opt-p -- optional parameter
p-obj -- positive object neg-n -- member of the nth negative set matched

Appendix E

Interface to the Dalle Molle Model

E.l The Original Dalle Molle Model

To run the Dalle Molle model, one had to enter these two functions, where the second function sets the values of the input parameters. Then the model was exectucted by calling the second function without any arguments. The model calls QSIM to do its simulation and then normally generates a visual display for humans rather than returning values.

```
;;;Edited by MURDOCK                3 Aug 94 19:18
;;;Edited by MURDOCK                3 Aug 94 20:33
;;;Edited by MURDOCK                3 Aug 94 22:05
(define-qde  hot-cold-mixing-tank
  (text "Revised model of a tank mixing a hot and a cold stream with
         redundant constraints")
  (quantity-spaces
    (volume (0 v* inf))
    (height (0 h* inf))           ;used for open-loop case
;   (height (0 h* h+ inf))        ;used for feed back control
    (outflow (0 of* inf))
    (inflow (0 if* inf))
    (netflow (mini 0 inf))
    (coldflow (0 fc- fc* fc+ inf)) ;used for open-loop case
;   (coldflow (0 fc* inf))
    (hotflow (0 fh- fh* fh+ inf))
    (tempout (0 tc* ts* th* inf))
    (coldtemp (0 tc* inf))
    (hottemp (0 th* inf))
    (DT (0 th*-tc* inf))
    (cf*DT (0 fc*DT inf))
    (hf*DT (0 fh*DT inf))
    (cf*DT/if (0 fc*DT/if* inf))
    (hf*DT/if (0 fh*DT/if* inf))
    (tempin (0 tc* ts* th* inf))
    (dT/dt (minf 0 inf))
    (ti-to (minf 0 inf))
    (if*ti-to (minf 0 inf))
; quantity spaces for p controller
;   (setheight (0 h- h* h+))
;   (errheight (mini 0 inf))
    )
```

```
  (constraints
    ((m+ volume height) (v* h*) (inf inf) (0 0))
    ((m+ outflow height) (of* h*) (inf inf) (0 0))
    ((add outflow netflow inflow) (of* 0 if*) (inf 0 inf) (0 0 0))
    ((add coldflow hotflow inflow) (fc* fh* if*) (0 0 0))
    ((d/dt volume netflow))
    ((add DT coldtemp hottemp) (th*-tc* tc* th*) (0 0 0))
    ((add tempin cf*DT/if hottemp) (th* 0 th*) (ts* fc*DT/if* th*) (0 0 0))
    ((add coldtemp hf*dt/if tempin) (tc* 0 tc*) (tc* fh*DT/if* ts*) (0 0 0))
    ((mult cf*DT/if inflow cf*DT) (fc*DT/if* if* fc*DT))
    ((mult hf*DT/if inflow hf*DT) (fh*DT/if* if* fh*DT))
    ((mult coldflow DT cf*DT) (fc* th*-tc* fc*DT))
    ((mult hotflow DT hf*DT) (fh* th*-tc* fh*DT))
    ((add tempout ti-to tempin) (th* 0 th*) (tc* 0 tc*) (ts* 0 ts*))
    ((d/dt tempout dT/dt))
    ((mult volume dT/dt if*ti-to) (v* 0 0))
    ((mult inflow ti-to if*ti-to) (if* 0 0))
; constraints for p controller
    ((add errheight height setheight) (0 h- h-) (0 h* h*) (0 h+ h+))
;   ((m+ coldflow errheight) (fc* 0))
    )

  (independent hottemp coldtemp coldflow hotflow)
; (independent hottemp coldtemp setheight hotflow)
  (history volume height outflow inflow netflow tempout tempin dT/dt
   DT cf*DT hf*DT cf*DT/if hf*DT/if ti-to if*ti-to)
; (history volume height outflow inflow netflow tempout tempin dT/dt
;          DT cf*DT hf*DT cf*DT/if hf*DT/if ti-to if*ti-to
;          err-height coldflow)
  (other (ignore-qdirs dT/dt))
; (other (ignore-qdirs dT/dt ti-to if*ti-to))
  (layout (height tempout) (netflow dT/dt)

  )


;;;Edited by MURDOCK                3 Aug 94 22:05
;;;Edited by MURDOCK                3 Aug 94 22:08
(defun hct-fc+ ()
  '(let ((normal nil)
(initial nil)
(analytic-functions-only t)
(*new-landmarks-at-initial-state*  nil))
    (declare (special normal initial analytic-functions-only))
    (setq *analytic-functions-only* t)
    (setq normal (make-initial-state  hot-cold-mixing-tank
 '((coldtemp (tc* std)) (hottemp (th* std))
   (hotflow (fh* std)) (coldflow (fc* std))
   (tempout (ts* std)) (height (h* std)))
 "Normal"))
    (setq  initial  (make-initial-state  hot-cold-mixing-tank
```

```
  '((coldtemp (tc* std)) (hottemp (th* std))
    (hotflow (fh* std)) (coldflow (fc+ std))
    (tempout (ts* nil)) (height (h* nil))
    )
  "Step increase in coldf low") )
    (qsim-display (qsim initial) :reference-states
  '((normal ,normal)))
    ))
```

E.2 The Macro that Generates the Dalle Molle Model

This macro was built to generate the functions that the Dalle Molle model normally required humans to enter. Notice the progn construct that contains the identical functions as above.

```
;;;Edited by MURDOCK              25 Oct 94 16:16
;;;Edited by MURDOCK              26 Oct 94 11:28
;;;Edited by MURDOCK              26 Oct 94 12:20
(defmacro simulate-mixing-tank (coldflow-before      ;before the step input
coldflow-after        ;after the step change
coldflow-q-space      ;list of 5, first is 0, last is inf
hotflow-before        ;before the step change
hotflow-after         ;after the step change
hotflow-q-space       ;list of 5, first is 0, last is inf
height
height-q-space        ;list of 3, first is 0, last is inf
tempout
tempout-q-space       ;list of 5, first is 0, last is inf
)
  (let* ((h* (second height-q-space))
 (fc- (second coldflow-q-space))
 (fc* (third coldflow-q-space))
 (fc+ (fourth coldflow-q-space))
 (fh- (second hotflow-q-space))
 (fh* (third hotflow-q-space))
 (fh+ (fourth hotflow-q-space))
 (tc* (second tempout-q-space))
 (ts* (third tempout-q-space))
 (th* (fourth tempout-q-space))
 (th*-tc* (make-concat-symbol (make-concat-symbol th* '-) tc*))
 (fc*DT (make-concat-symbol fc* 'DT))
 (fh*DT (make-concat-symbol fh* 'DT))
 (fc*DT/if* (make-concat-symbol fc*DT '/if*))
 (fh*DT/if* (make-concat-symbol fh*DT '/if*))
 (coldflow-b (car coldflow-before))
 (coldflow-a (car coldflow-after))
 (hotflow-b (car hotflow-before))
 (hotflow-a (car hotflow-after))
 (height-value (car height))
```

```
        (tempout-value (car tempout))

)


   '(progn
      (define-qde  hot-cold-mixing-tank
 (text "Revised model of tank mixing a hot and cold stream with
               redundant constraints.")
(quantity-spaces
  (volume (0 v* inf))
  (height (0 ,h* inf))
  (outflow (0 of* inf))
  (inflow (0 if* inf))
  (netflow (mini 0 inf))
  (coldflow (0 ,fc- ,fc* ,fc+ inf))
  (hotflow (0 ,fh- ,fh* ,fh+ inf))
  (tempout (0 ,tc* ,ts* ,th* inf))
  (coldtemp (0 ,tc* inf))
  (hottemp (0 ,th* inf))
  (DT (0 ,th*-tc* inf))
  (cf*DT (0 ,fc*DT inf))
  (hf*DT (0 ,fh*DT inf))
  (cf*DT/if (0 ,fc*DT/if* inf))
  (hf*DT/if (0 ,fh*DT/if* inf))
  (tempin (0 ,tc* ,ts* ,th* inf))
  (dT/dt (mini 0 inf))
  (ti-to (minf 0 inf))
  (if*ti-to (minf 0 inf))
  )

(constraints
  ((mt volume height) (v* ,h*) (inf inf) (0 0))
  ((mt outflow height) (of* ,h*) (inf inf) (0 0))
  ((add outflow netflow inflow) (of* 0 if*) (inf 0 inf) (0 0 0))
  ((add coldflow hotflow inflow) (,fc* ,fh* if*) (0 0 0))
  ((d/dt volume netflow))
  ((add DT coldtemp hottemp) (,th*-tc* ,tc* ,th*) (0 0 0))
  ((add tempin cf*DT/if hottemp) (,th* 0 ,th*) (,ts* ,fc*DT/if* ,th*) (0 0 0))
, ((add coldtemp hf*dt/if tempin) (,tc* 0 ,tc*) (,tc* ,fh*DT/if* ,ts*) (0 0 0))
  ((mult cf*DT/if inflow cf*DT) (,fc*DT/if* if* ,fc*DT))
  ((mult hf*DT/if inflow hf*DT) (,fh*DT/if* if* ,fh*DT))
  ((mult coldflow DT cf*DT) (,fc* ,th*-tc* ,fc*DT))
  ((mult hotflow DT hf*DT) (,fh* ,th*-tc* ,fh*DT))
  ((add tempout ti-to tempin) (,th* 0 ,th*) (,tc* 0 ,tc*) (,ts* 0 ,ts*))
  ((d/dt tempout dT/dt))
  ((mult volume dT/dt if*ti-to) (v* 0 0))
  ((mult inflow ti-to if*ti-to) (if* 0 0))
  )

(independent hottemp coldtemp coldflow hotflow)
(history volume height outflow inflow netflow tempout tempin dT/dt
```

```
 DT cf*DT hf*DT cf*DT/if hf*DT/if ti-to if*ti-to)
(other (ignore-qdirs dT/dt))
(layout (height tempout) (netflow dT/dt))
)

      (setq *analytic-functions-only* t)
      (setq *new-landmarks-at-initial-state* nil)

      (let ((normal (make-initial-state hot-cold-mixing-tank
              (quote
               ((coldtemp (,tc* std)) (hottemp (,th* std))
               (hotflow (,hotflow-b std)) (coldflow (,coldflow-b std))
               (tempout (,tempout-value std)) (height (height-value std))))
             "Normal"))
            (initial (make-initial-state hot-cold-mixing-tank
             (quote
              ((coldtemp (,tc* std)) (hottemp (,th* std))
              (hotflow (,hotflow-a std)) (coldflow (,coldflow-a std))
              (tempout (,tempout-value nil)) (height (,height-value nil))))
            "Step increase or decrease in either/both coldflow or hotflow."))
   )
(qsim initial)
(let ((final-states-list (delete-if #'(lambda (this-state)
(or (inconsistent-p this-state)
    (incomplete-p this-state)
    (null (state-status this-state))))
    (state-successors-leafs initial)))
      )
  (format t "~% Final states are: ~S" final-states-list)
  (cond ((and (eq (length final-states-list) 1)
      (quiescent-p (car final-states-list)))
 (let* ((final-state (car final-states-list))
(final-qvalues (state-qvalues final-state))

   (in-package 'user)
   (list (qsim::qmag (qsim::qval 'height final-state))
 (qsim::qspace 'height final-state)
 (qsim::qmag (qsim::qval 'tempout final-state))
 (qsim::qspace 'tempout final-state)
 (qsim::qmag (qsim::qval 'time final-state))
 (qsim::qspace 'time final-state))))
(t (in-package 'user)
   nil)))

      ))))
```

E.3 An Auxiliary Function that Retrieves Output

Since the Dalle Molle model and QSIM do not return values, we built this function to retrieve the values it generated rather than letting them be sent to a visual display intended for humans.

```
;;;Edited by MURDOCK                5 Aug 94 12:29
;;;Edited by MURDOCK                7 Oct 94 13:38
(defun state-successors-leafs (start-state)
  "The function STATE-SUCCESSORS-LEAFS takes the initial state generated by a
   QSIM simulation, held by the global variable *initial-state*, and returns
   a list of the final states generated in the simulation."

  (let ((new-queue (user::make-queue)))
    (user::enqueue new-queue start-state)
    (do ((search-queue  new-queue )
 (leaf-list  nil))

((user::empty-queue-p  search-queue)  leaf-list)

      (let ((succ-states
              (cdr (qsim:: state-successors
                     (user::queue-frontsearch-queue))))
    )
(if (null succ-states)
    (setq leaf-list (cons (user::queue-front search-queue) leaf-list))
    (mapcar #'(lambda (this-state) (user::enqueue search-queue  this-state))
    succ-states))
(user::dequeuesearch-queue)  ;Since dequeue works by side-effect,
                             ;it can't be used as the stepper form for
                             ;the SEARCH-QUEUE variable of the DO.
))))
```

E.4 An Auxiliary Function to Call the Macro

Since our system required models to be executed only by *function* call, we wrote this function as a wrapper for the macro above.

```
(defun  dalle-molle-simulation  (coldflow-before
coldflow-after
coldflow-q-space
hotflow-before
hotflow-after
hotflow-q-space
height
height-q-space
t empout
tempout-q-space
)
```

```
      (user::load "sys:site;nq-92.translations")
      (user:: make-system 'nq-92)
      (load"x23:murdock.models;dalle-molle-interface.lisp")

 (eval (funcall #'macroexpand
'(qsim::simulate-mixing-tank
   ,coldflow-before
   ,coldflow-after
   ,coldflow-q-space
   ,hotflow-before
   ,hotflow-after
   ,hotflow-q-space
   ,height
   ,height-q-space
   ,tempout
   ,tempout-q-space))
))
```

Appendix F

Proof of NP-Completeness for the Matching Problem

F.1 Formalizing the Model-Matching Problem

To make the problem of matching a model description to an equipment description amenable to formal analysis, this section introduces some representational modifications that preserve the original problem content. Some polynomial time operations to select parts of the equipment description graph produce a subgraph used as input for the formalized model matching problem. Since the following section shows the formalized model matching problem to be NP-complete, these polynomial subgraph selections do not significantly impact the complexity of the whole problem.

F.1.1 The Informal Input

Recall from Chapter 4 that type hierarchies (materials, geometries, functional types, etc.) form a vocabulary that describes both models and the equipment being diagnosed. Objects to which models may be applied, physical parts of the equipment, and parameters are represented by frame-like objects with links indicating which types they exemplify. Parameter objects are linked to the object they describe and have a set of attributes which may take on various types of values. Objects may also have links between them indicating physical connections in the equipment or other relationships. This representation scheme uses 13 types of links (and their inverses) to capture these possible relationships.

An equipment description is a set of objects each with at least one link to the type hierarchies and possibly having links chosen from the 13 link types indicating relationships between them. Some of these objects represent parameters and thus have a particular set of attributes, each of which may take on various types of values.

A model description contains objects (some of them being parameters) and links as in an equipment description, but here these objects represent what equipment the model may be applied to rather than representing a physically existing system as the equipment description does. Varying levels of abstraction in the model description allow it to be applied to more or less situations. The parameter objects represent what parameters are involved in the model's calculation, either as applicability conditions or as input or output. Besides the model and equipment descriptions, an equipment parameter and a matching model parameter are assumed to be given. Since these

two objects are required to match, they act as an anchored starting point for the problem. (The equipment parameter and model parameter correspond to the input to the M-R portion of the algorithms in this thesis, as described in Chapter 5.)

F.l.2 The Informal Problem

A model matches part of an equipment description when all of the model objects (possibly excluding parameters) have corresponding equipment objects of the same type as the model object or a subtype of the model object's type and the links between the model's objects have corresponding links between the equipment objects. In other words, objects match when the model object is more abstract or equally abstract as the equipment object, and links match only if they have the same name. For the problem at hand we assume we have one object in the model matched and anchored to one object in the equipment.

F.1.3 The Formal Input

Since an equipment description may be large and mostly unrelated to the model at hand, irrelevant parts may be pruned as follows. Notice that a model description's objects and an equipment description's objects are connected indirectly through the type hierarchies, since all objects must have at least one link to a type hierarchy object. First traverse all the EXEMPLIFIES links from the model objects to find thier associated type objects (called "model's types") in the type hierarchy. Only parts of the equipment description that are objects of these types or of subtypes of the model's types may match the model. By following the trees rooted at the model's types and made up of CAN-BE-A and EXEMPLIFIED-BY links, we reach all the correct type objects in the equipment description. For the purposes of formal analysis, each path to the leaves of these trees is replaced by a single EXEMPLIFIES link from the leaf to the root. If we then exclude all other objects not linked to our tree roots and include any links where the objects at both ends are in the selected set of objects, the resulting graph contains no objects of irrelevant types. This operation is a tree search so will require time polynomial in the number of nodes in the trees.

In the matching procedure, failure occurs because of either lack of objects of appropriate type, lack of correct links between objects, or unsatisfied value requirements for the attributes of parameter objects. Since parameter objects have a fixed number of attributes (See Section ref?.) which may take values from a fixed set of value types, then checking a model's requirements on a parameter's attributes requires constant time (assuming that we limit the representation of large numbers to some fixed number of significant digits.) Since the previously described tree search to find equipment objects exemplifying model types is polynomial, ignoring these constant time attribute checking operations will not affect the order of complexity in the final analysis.

The links between objects may also be simplified for analysis, since, except for the 5 dis-aggregation links (has-part, has-component, has-phase-component, has-spatial-portion, and has-instance), the link names are redundant with the types of the objects they connect. The representation method defined in this thesis uses only 13 links for use in the type hierarchies, the

equipment descriptions, and the model descriptions, each of which can only originate and terminate at particular types of objects. For example, the STREAH-OF-PORT link may only orignate at an object that exemplifies some port type and may only terminate at some object the exemplifies some stream type. For the purposes of this proof, we exclude descriptions that use the 5 disaggregation links. Some models that we have implemented, such as the Dittus-Boelter model (Section C.2), do not use these links. If these links happen to reduce the complexity of matching from NP-complete to polynomial, but problems that do not use them can be shown to be NP-complete, then the general problem is still NP-complete. So if we exclude the disaggregation links, any link name could be determined unambiguously from the types of objects at either end. For the purposes of this proof, we can treat links as unlabeled edges.

The simplifications described result in a graph G = ***(V, E)*** where the vertices V are partitioned into three subsets, the set of model description objects X, the set of type hierarchy objects C', and the set of equipment description objects U'. The set of edges E, made up of pairs of vertices from ***V,*** (v_i, v_j), have either both $v_i, v_j \in$ X, both $v_i, v_j \in$ U', $v_i \in$ X and $v_j \in C'$, or $v_i \in U'$ and $v_j \in C'$. G contains no edges with one vertex in X and one in U'. In other words, no model description objects are linked directly to equipment description objects. The graph $G = (X \cup C' \cup U', E)$ is the input for the formalized model matching problem, as defined below.

F.1.4 The Formal Problem

To match a model, a subset U'_{sub} of U' must "correspond" to X, and a subgraph containing the vertices of U'_{sub} must be isomorphic to the graph on X. U'_{sub} is said to ***correspond*** to X if there is a one-to-one correspondence between the elements in X and the elements in ***U'*** where for each x_i in X, the corresponding u'_i in U'_{sub} has edges to all the same $c_1, c_2, \ldots c_n$ in C' as x_i. We now define MODEL MATCHING to be the formalized version of the original problem.

Definition F.1 (MODEL MATCHING) ***Given a graph G = (X*** $\cup C' \cup$ ***U', E) where no edge in E is incident on both a vertex*** x_i ***in X and*** a ***vertex*** u'_i ***in*** U' ***and where X,*** C', ***and*** U' ***are disjoint sets*** *of* ***vertices, is there a*** *subgraph* ***on a subset*** *of* U' ***that corresponds and is isomorphic to the*** *subgraph* ***on X ?***

This definition states MODEL MATCHING as a decision problem so that the techniques for proving NP-completeness may be applied.

F.2 Complexity of Model-Matching

In this section, MODEL MATCHING is shown to be NP-complete according to the proof procedure described by [30]. First MODEL MATCHING is shown to be in the class NP. A transformation from an instance of a known NP-complete problem, The Satisifiability Problem (SAT), to an instance of MODEL MATCHING is given and shown to require polynomial time. The transformation is shown to produce an instance of MODEL MATCHING that has a match if and only if the instance of

SAT had a solution. This sequence of proofs shows that MODEL MATCHING is at least as hard as SAT, and if P $\neq$ ***NP*** then MODEL MATCHING is NP-complete.

F.2.1 MODEL-MATCHING is in NP

MODEL MATCHING $\in$ NP if a nondeterministic algorithm may guess a subset of vertices in U' and edges amongst those vertices as a possible solution to MODEL MATCHING and then verify in polynomial time that the guess is a solution. Since verification requires a comparison between a graph on the vertices in X and C' to a subgraph on vertices in U' and C', the complexity is $O(|X| + |C'|)^2$, which is clearly polynomial in the size of the MODEL MATCHING instance. **MODEL MATCHING** is therefore in NP.

F.2.2 Transformation from SAT to MODEL-MATCHING

To show that MODEL MATCHING is NP-complete, we provide an algorithm to transform an arbitrary instance of a known NP-complete problem, the Satisfiability Problem (SAT), into MODEL MATCHING. SAT was proved NP-complete by Cook [9] and was the first such known problem. In Section F.2.3 we demonstrate that this transformation can be performed in polynomial time.

Let $U = \{u_1, u_2, \ldots, u_n\}$ be a set of Boolean variables. We can then define ***truth assignment, literal,*** *clause*, and SAT as in [30].

Definition F.2 (truth assignment) ***A truth assignment for*** U ***is a function*** $t : U \rightarrow$ ***{T, F}.***

If $t(u) =$ ***T,*** then u is said to be "true" under t, and if ***t(u) = F*** then we say u is "false."

Definition F.3 (literal) ***For*** $u \in U$, ***u and*** $\overline{u}$ are ***literals*** **over** ***U. A*** **literal** ***u is true under truth assignment t if and only if the*** *variable* ***u is true under t. The literal ii is true under t if and only if the variable u is false.***

Definition F.4 (clause) ***A clause over*** U ***is a set of literals over*** U ***representing the disjunction of those literals.***

A clause is said to be satisfied under a truth assignment t if at least one of the literals in the clause is true under t.

Definition F.5 (The Satisfiability Problem-SAT) ***Given a set*** U ***of variables and a collection C of clauses over*** U, ***is there a truth assignment that satisfies all the clauses in C?***

We must construct a graph, G = (X UC' U ***U', E)*** where G has a subgraph among the vertices in ***U'*** that matches the graph on the vertices in X and corresponds through the vertices in C' if and only if the instance of SAT S = (U, ***C)*** from which we built G has a satisfying truth assignment. For each clause c_j in C, the graph G has a vertex c'_j forming the required set C' of vertices. For each clause c_j, there is a vertex x_j forming the set X. Exactly $|C|$ edges connect the vertices in X and the vertices from C' so that each edge covers one vertex from each set, and no vertex from

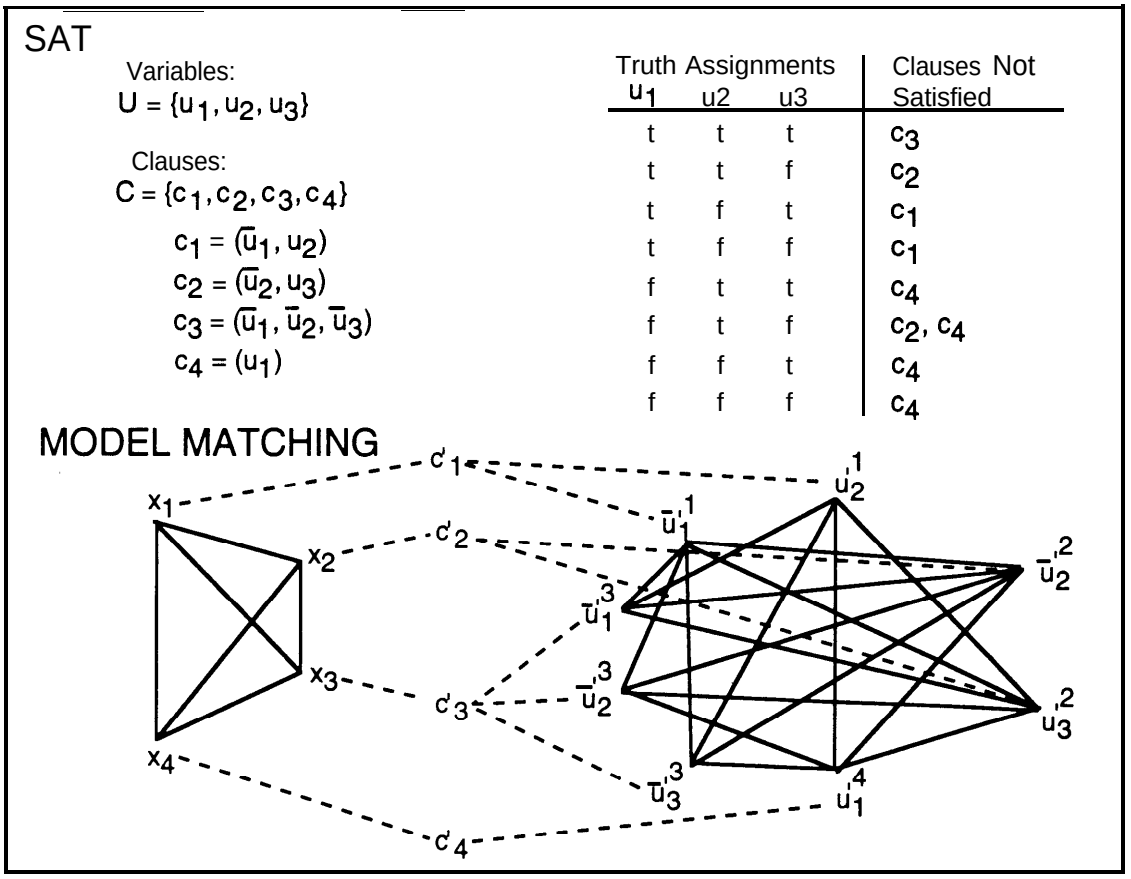


Figure F. 1: Tranformation of an instance of SAT that has no satisfying truth assignment to an instance of MODEL MATCHING that has no match.

either set is connected to more than one vertex in the other set. The edges (x_j, c'_j) determine correspondence in the instance of model matching. The vertices in X each have edges to every other vertex x_i in X. The vertices in X and the edges (x_i, x_j) constitute the model in MODEL MATCHING. See Figures F.l and F.2 for examples.

For each appearance of a literal in a clause of SAT, the graph G contains a vertex $u_i'^k$ (or $\overline{u}_i'^k$) forming the vertex set U'. The subscript indicates which variable the vertex represents and the superscript indicates in which clause the literal appears, since the same literal may appear in more than one clause. Edges (u'^k_i, c'_j) (or $(\overline{u}'^k_i, c'_j)$) connect each u'^k_i to exactly one vertex c'_j where $k = j$, representing the appearance of the literal u_i (or $\overline{u}_i$) in the clause c_j.

The graph G also contains edges among the vertices in U of the following forms:

1. (u'^k_i, u'^m_l) where $k \neq m$

   (Any two vertices representing positive literals have an edge between them unless they correspond to literal appearances in the same clause.)

2. $(\overline{u}'^k_i, u'^m_l)$ where $i \neq l$ and $k \neq m$

   (Any two vertices representing a positive literal and a negative literal have an edge between

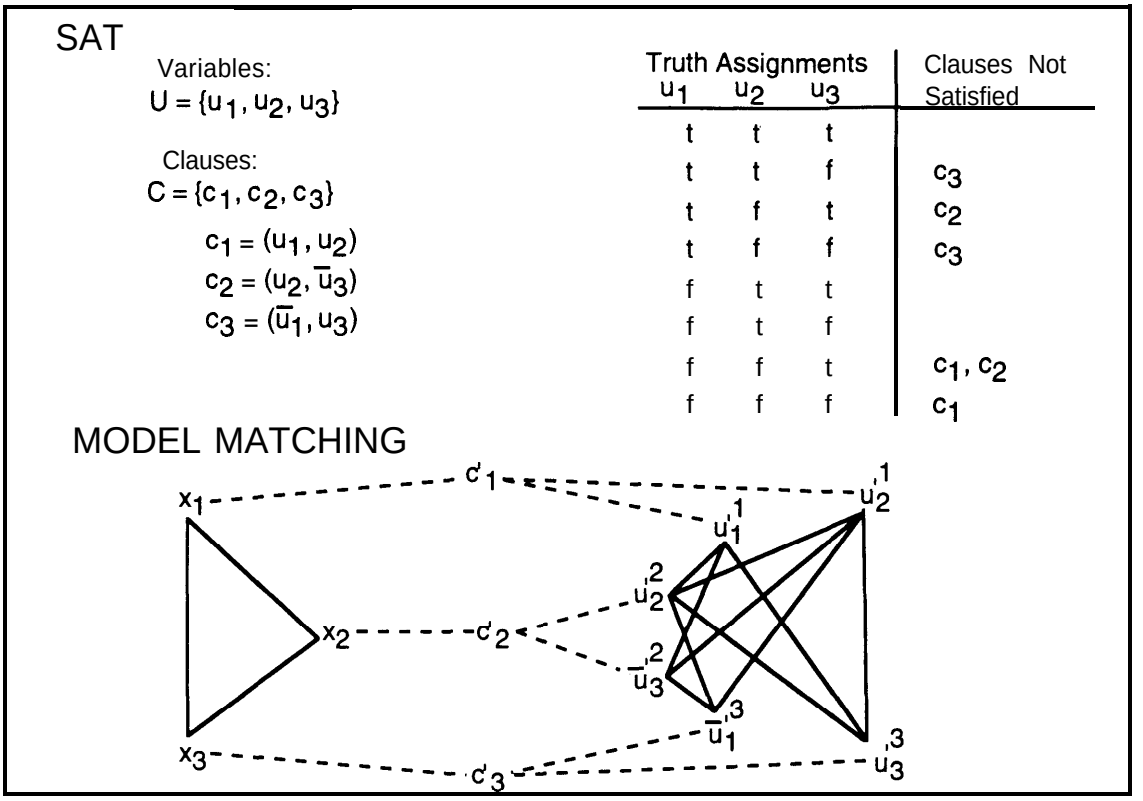


Figure F.2: Transformation of an instance of SAT that has three satisfying truth assignments to an instance of MODEL MATCHING that has at least one match for each satisfying truth assignment.

them unless they correspond to the same variable or unless they correspond to literal appearances in the same clause.)

3. $(\overline{u}_i'^k, \overline{u}_l'^m)$ where $k \neq m$
(Any two vertices representing negative literals have an edge between them unless they correspond to literal appearances in the same clause.)

When the SAT instance has no satisfying truth assignments, the MODEL MATCHING instance will have no matches as illustrated in Figure F.l. When the SAT instance has a satisfying truth assignment, this transformation produces instances of MODEL MATCHING that have at least as many matches as the originating instance of SAT had satisfying truth assignments. Figure F.2 shows an example with three satisfying truth assignments and four different matches. The additional matches result from the different possible ways that a satisfying literal may be chosen from the literals in a clause. From the example in Figure F.2 for the single truth assignment $(u_1 = T, u_2 = T, u_3 = T)$, we may choose the satisfying literals from the three clauses to be either u_1, u_2, us or u_2, u_2, u_3.

F.2.3 Transformation is Polynomial

Let the size of an instance S = **(U,** C) of SAT be measured by the number of of literals, u_i or $\overline{u}_i$, that may be formed from the variables u_i of U and the number of clauses c_j in C. If n is the number of variables and m the number of clauses, than the size of SAT can be no more than **2nm.** To generate the MODEL MATCHING graph, each of the following steps were performed:

1. Generate the vertices in C', one for each clause in C.
   (m vertices)

2. Generate the vertices in X, one for each clause in C.
   (m vertices)

3. Generate the edges between corresponding vertices in X and C'.
   (m edges)

4. Generate edges between every pair of vertices in X.
   ($(m-1)m/2$ edges)

5. Generate vertices in **U';** one for each appearance of a literal in a clause.
   (2nm vertices maximum)

6. Generate edges from each vertex $u_i'^k$ or $\overline{u}_i'^k$ in **U'** to the vertex c_k' in C' which represents its clause.
   (2nm edges maximum)

7. Generate edges between vertices in **U'** according to the criteria given in the transformation.
   ($2nm(2nm+1)/2$ operations maximum, since each pair of vertices in **U'** would be compared to the criteria to determine if an edge should be placed there.)

The transformation is clearly polynomial, being composed of the sum of the polynomial operations described, with the most complex operation being $O(n^2m^2)$.

F.2.4 SAT is Satisfiable if and only if there is a Match

It 'remains to be shown that the clauses C of SAT are satisfiable if and only if the graph G = $(X \cup C' \cup U', E)$ has a subgraph on a subset of vertices in **U'** that matches and corresponds to the subgraph on vertices in X.

First, suppose that an instance of SAT, S = **(U, C),** has a satisfying truth assignment, $t : U \rightarrow$ **{T, F}.** Then at least one literal from each clause in C, either a variable u_i or its complement $\overline{u}_i$, must be assigned T. The graph G = (X UC'U **U'; E)** constructed from S has its vertices partitioned into three sets, where the sets X and C' each coutain one vertex for each clause in C and each x_i in X has exactly one edge to a vertex c_j' in C', and each c_j' has exactly one edge to a vertex x_i. In addition, each x_i has an edge to every other vertex in X, forming a clique of size $|C|$. **U'** contains a vertex $u_i'^k$ or $\overline{u}_i'^k$ for each literal appearing in each clause of C in SAT and each vertex in C' will

have an edge to the same number of vertices in U' as there were literals in the corresponding clause of the SAT problem. The vertices in U' each have one edge to a vertex in C'.

Since this instance of SAT has a satisfying truth assignment for its variables, then at least one literal from each clause is true, and that satisfying literal cannot be the complement of the satisfying literal of any other clause. Therefore, any vertex in U' that corresponds to a satisfying literal must have edges to all vertices corresponding to satisfying literals, forming a clique of at least size $|C|$. Furthermore, since a vertex $u_i'^k$ or $\overline{u}_i'^k$ in U' cannot have an edge to any other $u_j'^k$ or $\overline{u}_j'^k$ that has an edge to the same vertex c_k' in C', then the clique will be of exactly size $|C|$ and its vertices will each have an edge to a different vertex in C' corresponding to those edges from vertices in X to those in C'. This clique of vertices in U' having the required matching and correspondence properties to the clique in X is a MODEL MATCH.

Conversely, suppose that graph G = (X ∪ C' ∪ U', E) has a clique amongst the vertices in U' that corresponds to the clique containing all vertices in X, where the correspondence requires that each vertex of the clique in U' has exactly one edge to a vertex in C' and vice versa. This clique is of exactly size $|C|$, the number of clauses in the corresponding instance of SAT. Since no vertex $u_i'^k$ can have an edge to a vertex $\overline{u}_i'^l$, that represents the complementary literal, then the vertices of this clique form a truth assignment for this instance of SAT in the following way. Variables that are represented in the clique vertices as positive literals are assigned ***T,*** and those represented as negative literals in the clique vertices are assigned ***F.*** Variables not represented in the clique are assigned arbitrarily. Since the clique has each of its vertices attached to a different vertex in C', the set of vertices representing the clauses, the truth assignment contains one true literal for each clause, and thus is a satisfying truth assignment for this instance of SAT.

Bibliography

[1] Sanjaya Addanki, Roberto Cremonini, and J. Scott Penberthy. Graphs of models. *Artificial Intelligence,* 51:145–177, 1991.

[2] R. Byron Bird, Warren E. Stewart, and Edwin N. Lightfoot. *Transport Phenomena.* John Wiley & Sons, Inc., New York, NY, 1960.

[3] Andrea Bonarini and Piera Sassaroli. Opportunistic multimodel-based diagnosis: framing all the knowledge we have to diagnose complex artifacts. In *Proceedings of the Ninth Conference on Artificial Intelligence for Applications,* pages 429-436. Los Alamitos, CA, 1993. IEEE Computer Society Press.

[4] Scott Bublin and R. L. Kashyap. Generating fault hypotheses with a functional model in machine-fault diagnosis. *Applied Artificial Intelligence,* 6:353–382, 1992.

[5] Tom Bylander and B. Chandrasekaran. Understanding behavior using consolidation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence,* pages 450-454, Los Altos, CA, 1985. Morgan Kaufmann.

[6] Nicholas P. Cheremisinoff and Paul N. Cheremisinoff. *Pumps and Pumping Operations.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[7] John W. Collins and Kenneth D. Forbus. Reasoning about fluids via molecular collections. In *Proceedings of the Sixth National Conference on Artificial Intelligence,* pages 503-507, Los Altos, CA, 1987. Morgan Kaufmann.

[8] Luca Console and Pietro Torasso. A spectrum of logical definitions of model-based diagnosis. In Walter Hamscher, Luca Console, and Johan de Kleer, editors, *Readings in Model-Based Diagnosis.* Morgan Kaufmann, San Mateo, CA, 1992.

[9] Stephen A. Cook. The complexity of theorem proving procedures. In *3rd Annual ACM Symposium on Theory of Computing,* pages 151-158, New York, 1971. Association for Computing Machinery.

[10] Philippe Dague, Olivier Raiman, and Philippe Devès. Troubleshooting: when modeling is the trouble. In *Proceedings of the Sixth National Conference on Artificial Intelligence,* pages 600-605, Los Altos, CA, 1987. Morgan Kaufmann.

[11] David Dalle Molle. *Qualitative Simulation of* Dynamic *Chemical Processes.* PhD thesis, University of Texas at Austin, August 1989. Published as technical report AI-89-107.

[12] Ernest Davis. Order of magnitude reasoning in qualitative differential equations. In D. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann, San Mateo, CA, 1990.

[13] Randall Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence,* 24(l-3):347-410, December 1984.

[14] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence,* 28:127–163, 1986.

[15] Johan de Kleer. Focusing on probable diagnoses. In *Proceedings of the Ninth National Conference on Artificial Intelligence,* pages 842-848, Menlo Park, CA, 1991. AAAI Press/MIT Press.

[16] Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artificial Intelligence,* 56:197–222, 1992.

[17] Johan de Kleer, Olivier Raiman, and Mark Shirley. One step lookahead is pretty good. In Walter Hamscher, Luca Console, and Johan de Kleer, editors, *Readings in Model-Based Diagnosis,* pages 138-142. Morgan Kaufmann, San Mateo, CA, 1992.

[18] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence,* 32:97–130, 1987.

[19] Johan de Kleer and Brian C. Williams. Diagnosis with behavioral modes. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* pages 1324-1330, San Mateo, CA, 1989. Morgan Kaufmann.

[20] J. Patrick Dishaw and Jeff Y.-C. Pan. AESOP: a simulation-based knowledge system for CMOS process diagnosis. *IEEE Transactions on Semiconductor Manufacturing,* 2:94–103, 1989.

[21] Jon Doyle. A truth maintenance system. *Artificial Intelligence,* 12:231–272, 1979.

[22] John W. Duf our and William E. Nelson. *Centrifugal Pump Sourcebook.* McGraw-Hill, New York, 1992.

[23] Brian Falkenhainer and Kenneth D. Forbus. Setting up large scale qualitative models. In *Proceedings of the Seventh National Conference on Artificial Intelligence,* pages 301-306, San Mateo, CA, 1988. Morgan Kaufmann.

[24] Brian Falkenhainer and Kenneth D. Forbus. Compositional modeling: finding the right model for the job. *Artificial Intelligence,* 51(l-3):95-143, October 1991.

[25] Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence,* 24:85–168, 1984.

[26] Kenneth D. Forbus and Peter B. Whalley. Using qualitative physics to build articulate software for thermodynamics education. In *Proceedings of the Twelfth National Conference on Artificial Intelligence,* pages 1175-1182, Menlo Park, CA, 1994. AAAI Press/MIT Press.

[27] Gregory G. Freeman. *Application of Analytic Models to Computer-Aided Integrated Circuit Device Diagnosis and Parametric Test Selection.* PhD thesis, Stanford University, March 1991. Published by Stanford Electrical Engineering Department as technical report G834-2.

[28] K. Funakoshi and K. Mizuno. A rule-based VLSI process flow validation system with macroscopic process simulation. *IEEE Transactions on Semiconductor Manufacturing,* 3:239–246, 1990.

[29] Massimo Gallanti, Marco Roncato, and Alberto Stefanini. A diagnostic algorithm based on models at different levels of abstraction. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* pages 1350-1355, San Mateo, CA, 1989. Morgan Kaufmann.

[30] Michael R. Gareyand David S. Johnson. *Computers and Intractability.* W.H. Freeman and Company, New York, 1979.

[31] Michael R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence,* 24(l-3):411-436, December 1984.

[32] S. D. Grantham and L. H. Ungar. A first principles approach to automated troubleshooting of chemical plants. *Computers and Chemical Engineering,* 14:783–798, 1990.

[33] R. A. Greenkorn and D. P. Kessler. *Transfer Operations.* McGraw-Hill, Inc., New York, 1972.

[34] Walter C. Hamscher. Compositional modeling: finding the right model for the job. *Artificial Intelligence,* 51(1–3), October 1991.

[35] Patrick J. Hayes. Naive physics I: Ontology for liquids. In J.R. Hobbs and B.C. Moore, editors, *Formal Theories of the Commonsense World,* pages l-36. Ablex Publishing Corporation, Norwood, NJ, 1985.

[36] Barbara Hayes-Roth. A blackboard architecture for control. *Artificial Intelligence,* 26:251–321, 1985.

[37] Barbara Hayes-Roth. An architecture for adaptive intelligent systems. *To appear in Artificial Intelligence, 72, 1995.*

[38] Yumi Iwasaki and Alon Y. Levy. Automated model section for simulation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence,* pages 1183-1190, Menlo Park, CA, 1994. AAAI Press/MIT Press.

[39] Yumi Iwasaki and Herbert A. Simon. Causality in device behavior. *Artificial Intelligence,* 29:3–32, 1986.

[40] Kenneth Man kam Yip. Model simplification by asymptotic order of magnitude reasoning. In *Proceedings of the Eleventh Nataonal Conference on Artificial Intelligence,* pages 634-640, Menlo Park, CA, 1993. AAAI Press/MIT Press.

[41] M. A. Kramer. Malfunction diagnosis using quantitative models with non-boolean reasoning in expert systems. *AICHE Journal,* 33:130–140, 1987.

[42] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence,* 29:289–388, 1986.

[43] Douglas B. Lenat and R.V. Guha. *Building Large Knowledge-Based Systems: Rpresenatation and Inference in the Cyc Project.* Addison-Wesley Publishing Company, Inc., Reading, MA, 1990.

[44] Octave Levenspiel. *Chemical Reaction Engineering.* John Wiley & Sons, Inc., New York, NY, 1972.

[45] Zheng-Yang Liu. Integrating two ontologies for electronics. In Boi Faltings and Peter Struss, editors, *Recent Advances in Qualitative Physics,* pages 153-168. MIT Press, Cambridge, MA, 1992.

[46] Zheng-Yang Liu and Arthur M. Farley. Shifting ontological perspectives in reasoning about physical systems. In *Proceedings of the Eighth National Conference on Artificial Intelligence,* pages 395-400, Menlo Park, CA, 1990. AAAI Press/MIT Press.

[47] M. L. Mavrovouniotis and G. Stephanopoulos. Formal order-of-magnitude reasoning in process engineering. *Computers and Chemical Engineering,* 12:867–880, 1988.

[48] Michael Model1 and Robert C. Reid. *Thermodynamics and its Applications.* Prentice-Hall, Inc., Englewood Cliffs, NJ, second edition, 1983.

[49] John L. Mohammed. *Combining Experiential and Theoretical Knowledge in the Domain of Semiconductor Manufacturing Equipment.* PhD thesis, Stanford University, August 1994. Published by Stanford Computer Science Department as technical report STAN-CS-94-1526, also KSL-94-62.

[50] John L. Mohammed and Reid G. Simmons. Qualitative simulation of semiconductor fabrication. In *Proceedings of the Fifth National Conference on Artificial Intelligence,* pages 794–799, Los Altos, CA, 1986. Morgan Kaufmann.

[51] P. Pandurang Nayak. *Automated Modeling of Physical Systems.* PhD thesis, Stanford University, September 1992. Published by Stanford Computer Science Department as technical report STAN-CS-92-1443.

[52] P. Pandurang Nayak, Leo Joskowicz, and Sanjaya Addanki. Automated model section using context-dependent behaviors. In *Proceedings of the Tenth National Conference on Artificial Intelligence,* pages 710–716, Menlo Park, CA, 1992. AAAI Press/MIT Press.

[53] Hwee Tou Ng. Model-based, multiple fault diagnosis of time-varying, continuous physical systems. In *Proceedings of the Sixth Conference on Artificial Intelligence Applications,* pages 9-15. IEEE. 1990.

[54] 0. 0. Oyeleye, F. E. Finch, and M. A. Kramer. Qualitative modeling and fault diagnosis of dynamic processes by MIDAS. In Walter Hamscher, Luca Console, and Johan de Kleer, *editors, Readings in Model-Based Diagnosis,* pages 262-275. Morgan Kaufmann, San Mateo, CA, 1992.

[55] Robert H. Perry and Cecil H. Chilton. *Chemical Engineers' Handbook.* McGraw-Hill, Inc., New York, 1973.

[56] D. R. Pi'tts and L. E. Sissom. *Theory and Problems of Heat Transfer.* McGraw-Hill, Inc., New York, 1977.

[57] Olivier Raiman. Order of magnitude reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence,* pages 100-104, San Mateo, CA, 1988. Morgan Kaufmann.

[58] Olivier Raiman. The alibi principle. In Walter Hamscher, Luca Console, and Johan de Kleer, editors, *Readings in Model-Based Diagnosis,* pages 66-70. Morgan Kaufmann, San Mateo, CA, 1992.

[59] Olivier Raiman, Johan de Kleer, Vijay Saraswat, and Mark Shirley. Characterizing non-intermittent faults. In *Proceedings of the Ninth National Conference on Artificial Intelligence,* pages 849-854, Menlo Park, CA, 1991. AAAI Press/MIT Press.

[60] Shankar A. Rajamoney and San Hoe Koo. Qualitative reasoning with microscopic theories. In ***Proceedings** of **the Eighth National Conference on Artificial Intelligence,*** pages 401-406, Menlo Park, CA, 1990. AAAI Press/MIT Press.

[61] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence,* 32:57–95, 1987.

[62] S. H. Rich and V. Venkatasubramanian. Model-based reasoning in diagnostic expert systems for chemical process plants. ***Computers and Chemical Engineering,*** 11:111–122, 1987.

[63] Jeff Rickel and Bruce Porter. Automated modeling for answering prediction questions: Selecting the time scale and system boundary. In ***Proceedings** of **the Twelfth National Conference** on Artificial Intelligence,* pages 1191-1198, Menlo Park, CA, 1994. AAAI Press/MIT Press.

[64] Geoffrey W. Rutledge and Ross Shachter. A method for the dynamic selection of models. In P. Cheesman and R. W. Oldford, editors, ***Selecting Models from Data: AI and Statistics IV.*** Springer-Verlag, 1994.

[65] Sharad Saxena and Amy Unruh. Diagnosis of semiconductor manufacturing equipment and processes. ***IEEE Transactions on Semiconductor** Manufacturing,* 7:220–232, 1994.

[66] Ethan A. Scarl, John R. Jamieson, and Carl I. Delaune. A fault detection and isolation method applid to liquid oxygen for the space shuttle. In ***Proceedings** of **the Ninth International Joint Conference on Artificial Intelligence,*** pages 414-416, Los Altos, CA, 1985. Morgan Kaufmann.