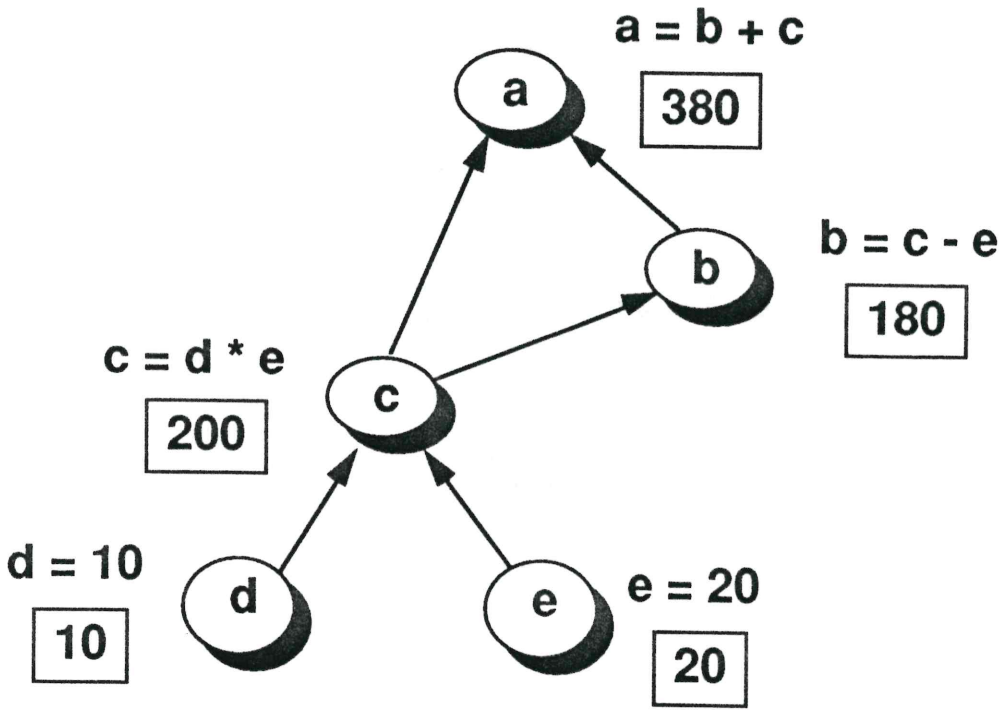
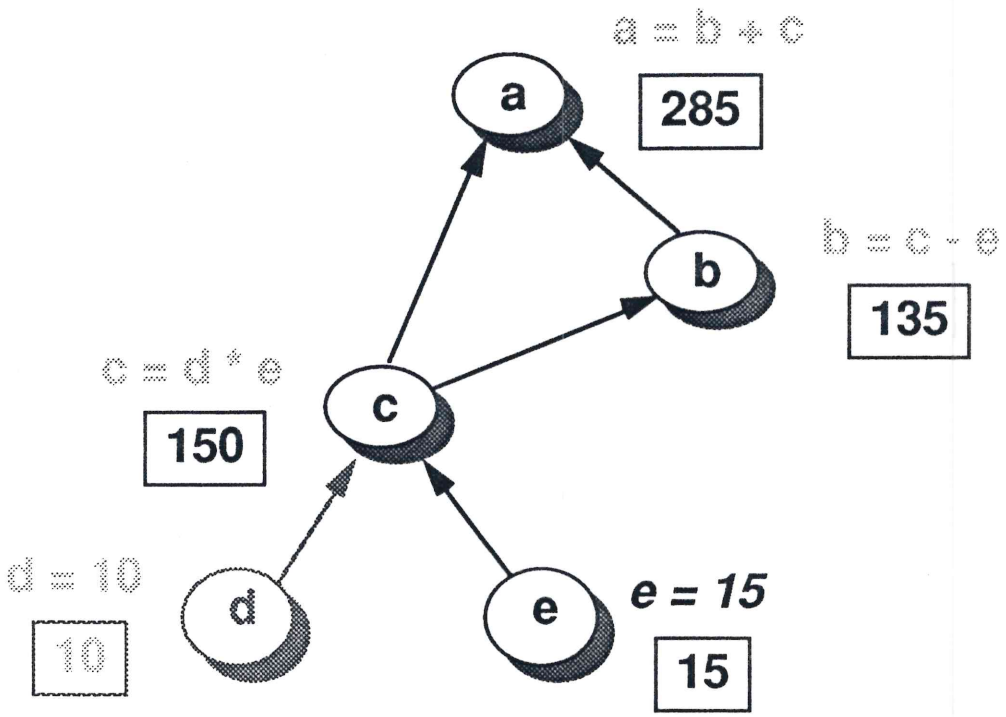
Definition Management



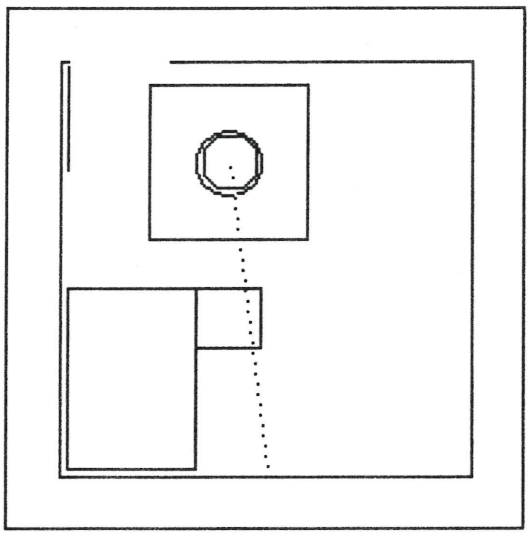
Definition Management



CAD DEMO

Table obstructs door

Cable is not long enough



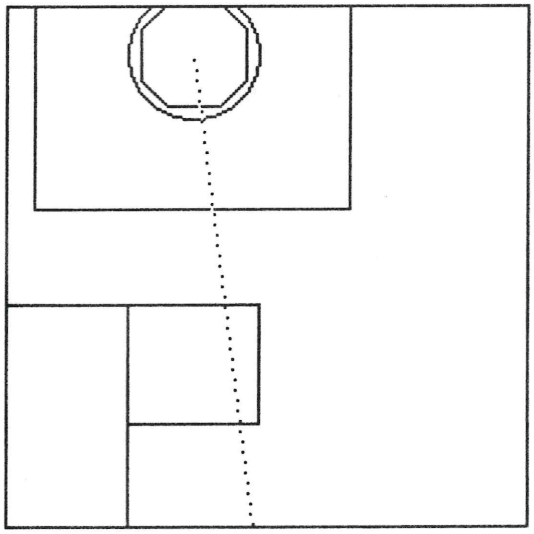


Table Position

1:Up

3:Left 4:Right

2:Down

9:Use Plug 2

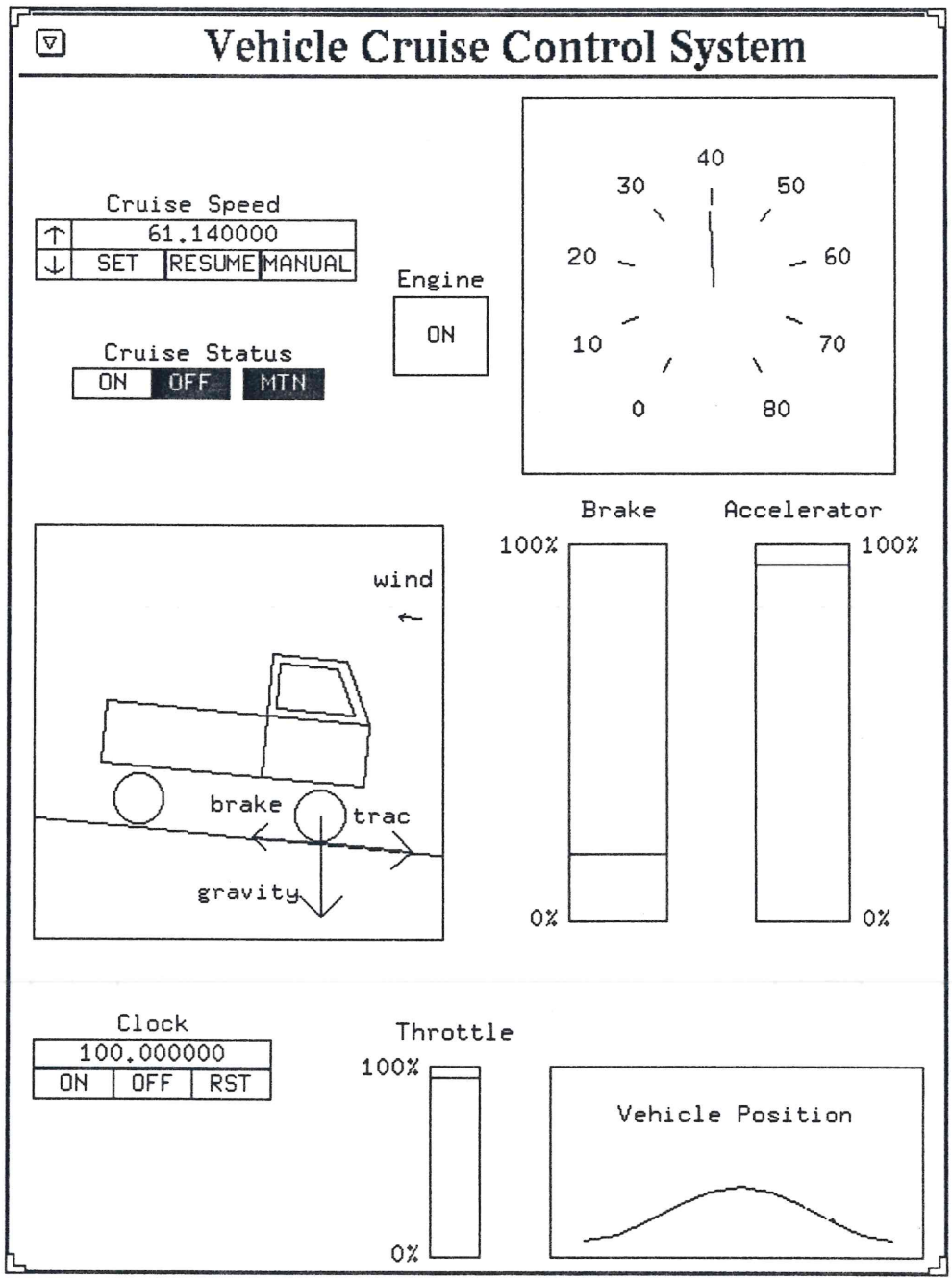
10:Close Door

Zoom Position

5:Up

7:Left 8:Right

6:Down



Sample Output of the Vehicle Cruise Control Simulation

Implementing Definitive Notations

1. Set Naming Scheme

DoNaLD Name	EDEN Name
table	_table
table/drawer	_table_drawer
table/drawer/width	_table_drawer_width

2. Implement Data Types & Operators

DoNaLD Type	EDEN Type
integer	integer
point	['C', integer, integer]
line	['L', point, point]

DoNaLD Operator	EDEN Operator/ Function
div	/

+ (vector sum)	func vector_add {
	para p1, p2;
	return [
	'C',
	p1[1] + p2[1],
	p1[2] + p2[2]
	];
	}

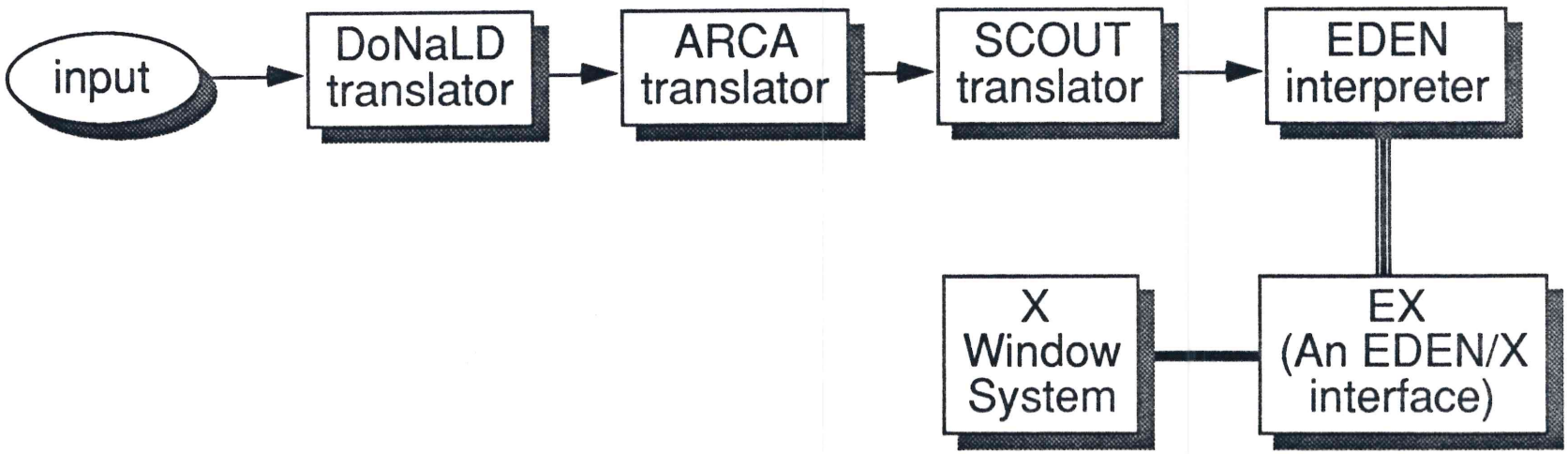
3. Implement Implicit Actions

DoNaLD Code	EDEN Action Specification
integer i	N/A

point p	proc P_p: _p {
	plot_point(&_p);
	}

line L	proc P_L: _L {
	plot_line(&_L);
	}

Run-time Structure of the Scout System

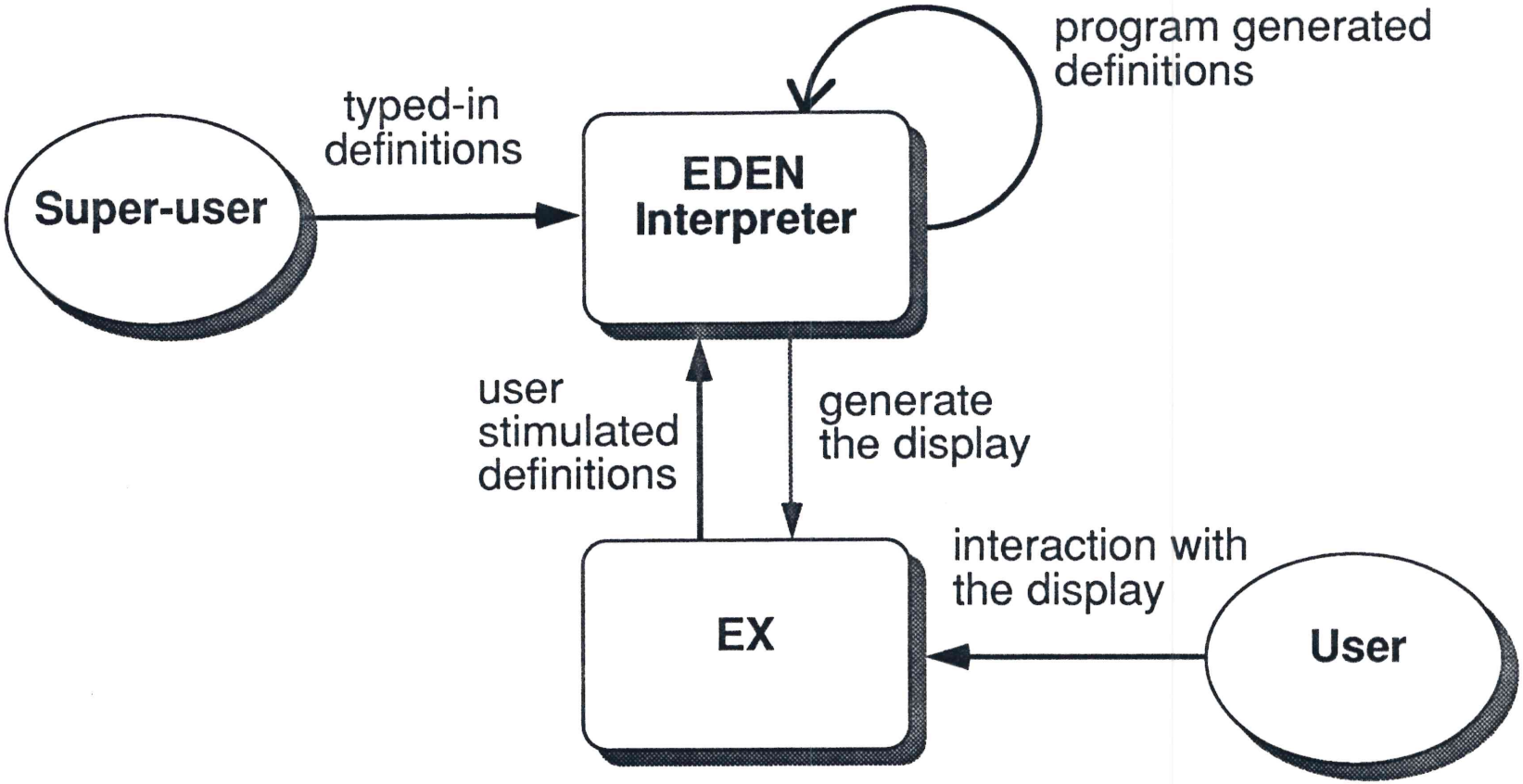


```
setBtn is (crSetBtn_mouse_1[2] == 4) ? pbDown : pbUp;
resBtn is (crResumeBtn_mouse_1[2] == 4) ? pbDown : pbUp;
manBtn is (crManualBtn_mouse_1[2] == 4) ? pbDown : pbUp;

%scout
window vehicle;
vehicle = {
     type: DONALD,
     box: [{10, 240}, {230, 460}],
     pict: "VEHICLE",
     xmin: -700,  ymin: -300,
     xmax: 300,   ymax: 700,
     border: 1
};

%donald
viewport MODEL
openshape conceptCar
within conceptCar {
     circle frontWheel, backWheel
     frontWheel = circle({0,0}, 60)
     backWheel = circle({-450, 0}, 60)
```

Extract From Vehicle Cruise Control Simulation



Sources of Definitions

```
%scout
S>      .
        .  definitions in Scout
        .
%donald
D[/]>   .
        .  definitions in DoNaLD
        .
%eden
        .
        .  EDEN definitions and statements
        .  (No prompt because of -n option)
        .
%error
        .
        .  EDEN as well
        .


%donald
    .
    .
    .
?   a line of EDEN

?A_cable = "linestyle=dashed,dash=13";
```

Attributes of DoNaLD lines

color	
linestyle	dotted, dashed, solid(default)
dash	default: 4 (the same as 44)
linewidth	default: 0

```
viewport MODEL              # not to be displayed

openshape conceptCar
within conceptCar {
        circle frontWheel, backwheel
        frontWheel = circle({0,0}, 60)
        ...
}

viewport VEHICLE            # displayed in VCCS

shape vehicle
vehicle = rot(conceptCar, {0,0}, gradient)
```

```
window brakePedal;
brakePedal  = {
   type:        DONALD
   box:         [brakeOrg,
                 brakeOrg + {BAwidth, BALength}]
   pict:        "BRAKE"
   xmax:        100
   ymax:        100
   border:      1
   sensitive:   ON
};

window brakeMark1;
brakeMark1 = {
   string:      "100%"
   frame:       ([brakePedal.box.nw-{5.c,1.r/2}, 1, 4])
   alignment: RIGHT
};




brakePedal_mouse = [1, 4, 1, 37, 50];


brakeMark1_mouse_1 = [1, 4, 0, 450, 600];
```

```
chime = 0;

proc clock_watcher : clock {
        if (clock - clock_init) >= 5) {
                clock_init = clock;
                chime++;
        }
        SendToEden("clock = "//str(time())//";\n");
}

proc SendToEden { para message;
        auto ok;
        ok = send_msg(stdmsg, [3, message], 0);
        if (ok == -1)
                error("can't write to message queue");
}

proc bell : chime { if (chime) writeln("Bell!"); }

clock = clock_init = time();
```

Advanced EDEN Features

? _cablelength;

eager();

nameof(&var);

` string_expr `

execute(string_expr);

forget(variable_name);