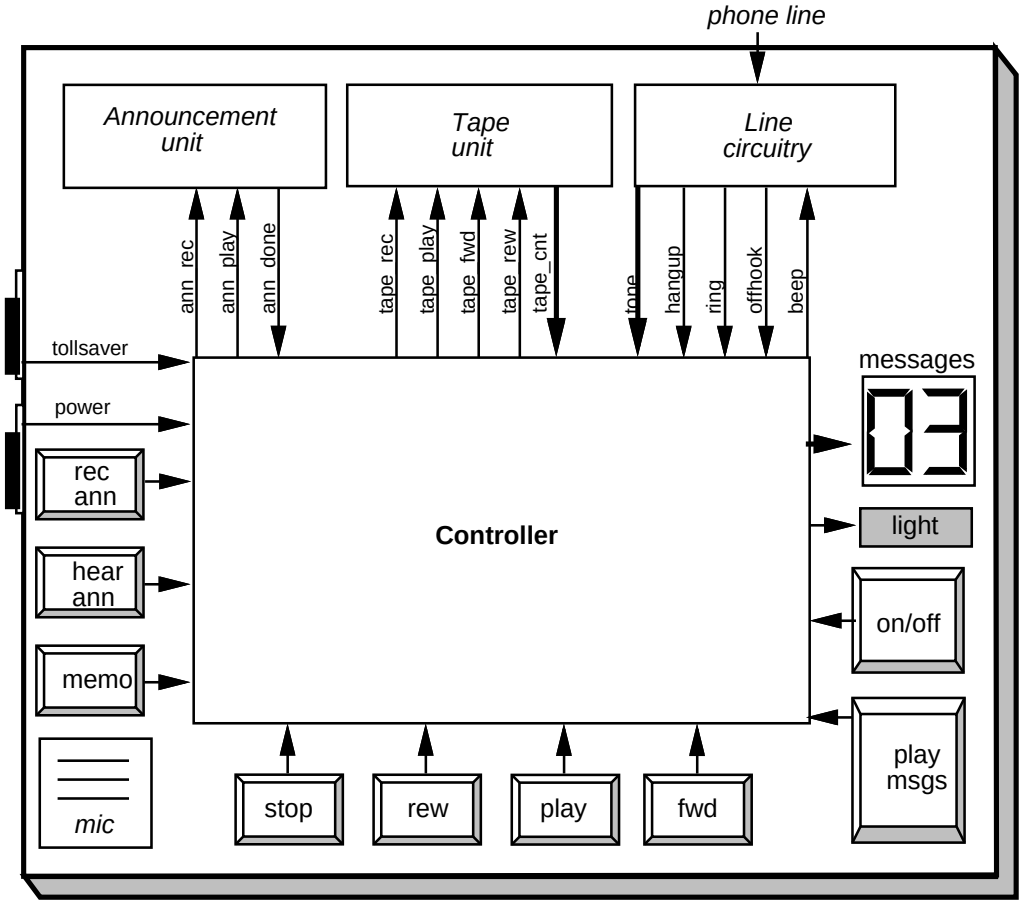
Specification example

- An executable specification-language enables:
  - Early verification
  - Precision
  - Automation
  - Documentation
- A good language/model match reduces:
  - Capture time
  - Comprehension time
  - Functional errors

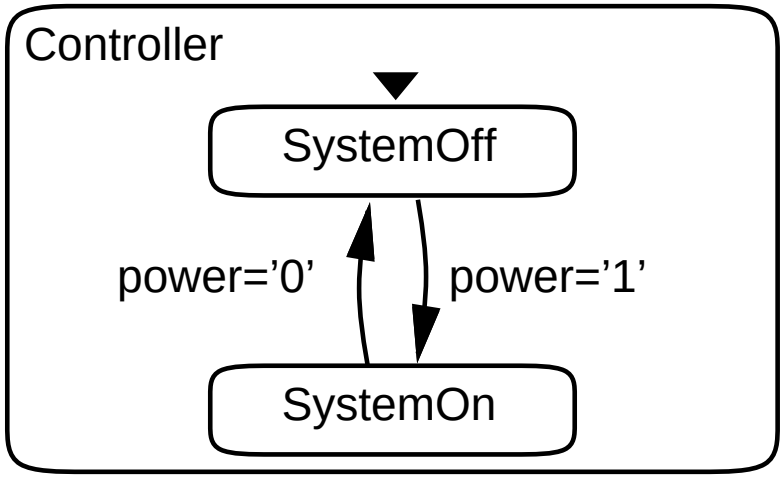
Outline

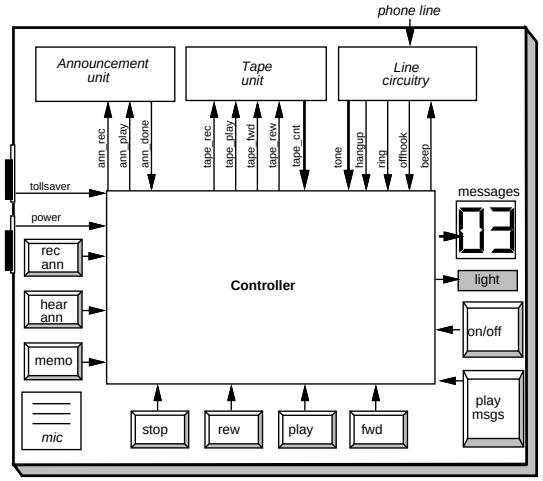
- Capture an example's model in a particular language
  - PSM model in the SpecCharts language
- Point out the benefits of a good language/model match
- Highlight experiments that demonstrate those benefits

Answering machine controller's environment



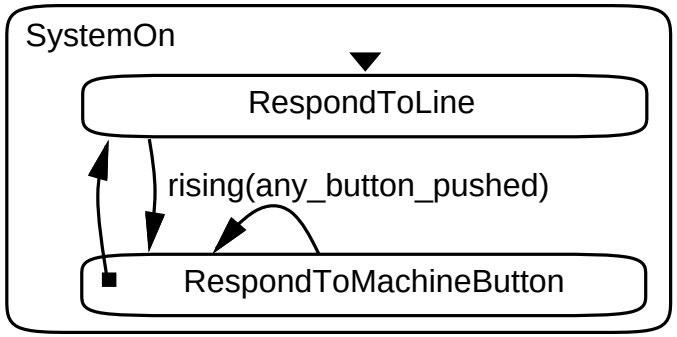
Highest-level view of the controller

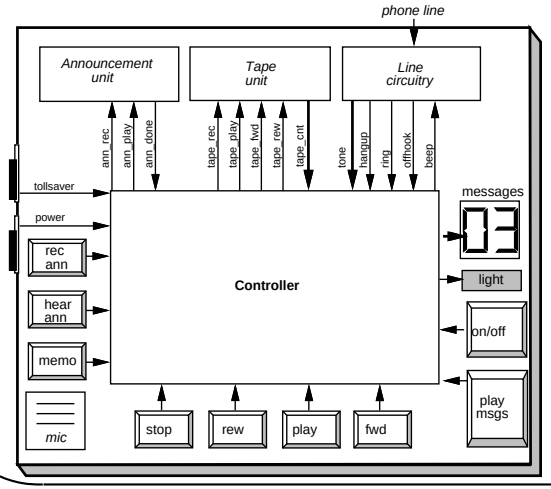




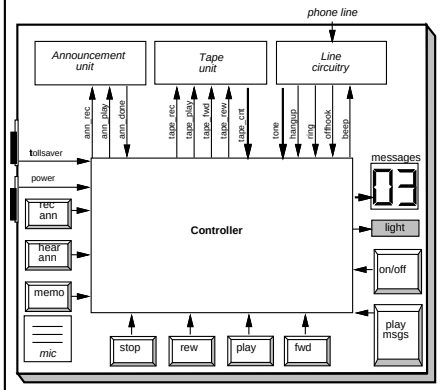
The SystemOn behavior

- System usually responds to the line
- Pressing any machine button gets immediate response



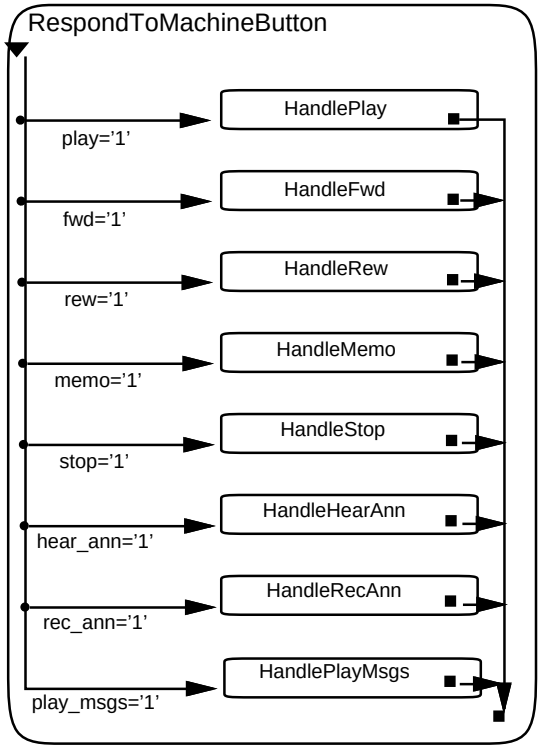


The RespondToMachineButton behavior




```
behavior  RespondToMachineButton
  type code is
begin
   if (play='1') then
      HandlePlay;
   elsif (fwd='1') then
      HandleFwd;
   elsif (rew='1') then
      HandleRew;
   elsif (memo='1') then
      HandleMemo;
   elsif (stop='1') then
      HandleStop;
   elsif (hear_ann='1')  then
      HandleHearAnn;
   elsif (rec_ann='1') then
      HandleRecAnn;
   elsif (play_msgs='1') then
      HandlePlayMsgs;
   end if;
end;
```

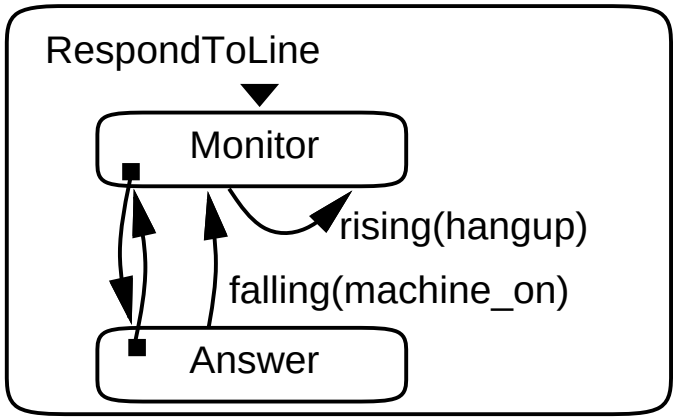
(a)

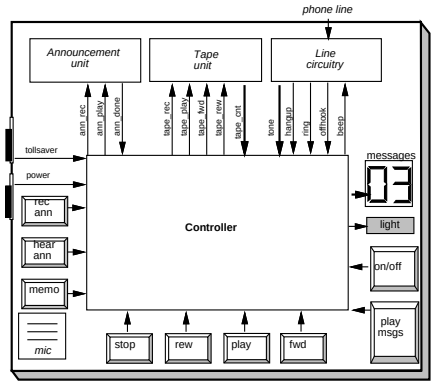


(b)

The RespondToLine behavior

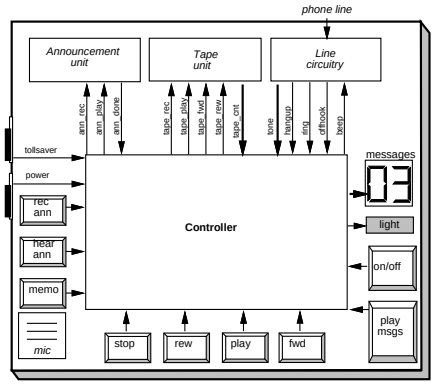
- Monitors line for rings
- Answers line
- Responds to exceptions
  - Hangup
  - Machine turned off





The Monitor behavior

- Counts for required rings
- Requirements may change



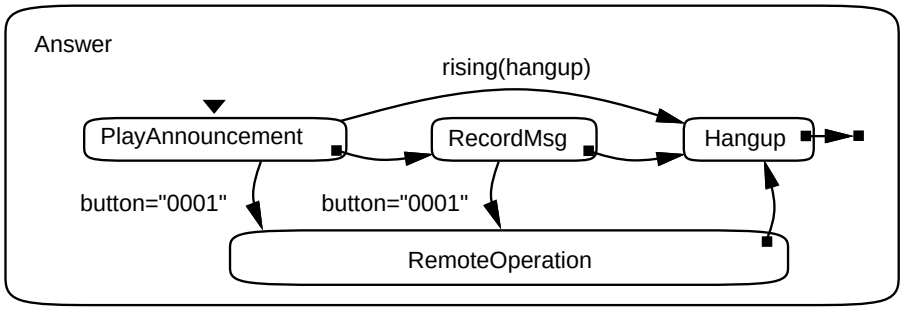

```
Monitor
        signal  rings_to_wait : integer range 1 to 20 := 4;
        function  DetermineRingsToWait return integer is begin
            if ((num_msgs > 0) and (tollsaver='1')  and (machine_on='1')) then
               return(2);
            elsif (machine_on='1') then
               return(4);
            else
               return(15);
            end if;
        end;
```

```
MaintainRingsToWait                          CountRings
                                                variable I : integer range 0 to 20;

loop                                            i := 0;
   rings_to_wait <= DetermineRingsToWait;       while (i < rings_to_wait) loop
   wait on tollsaver, machine_on;                  wait on rings_to_wait, ring;
end loop;                                          if (rising(ring)) then
                                                      i := i + 1;
                                                   end if;
                                                end loop;
```


UC Irvine

The Answer behavior



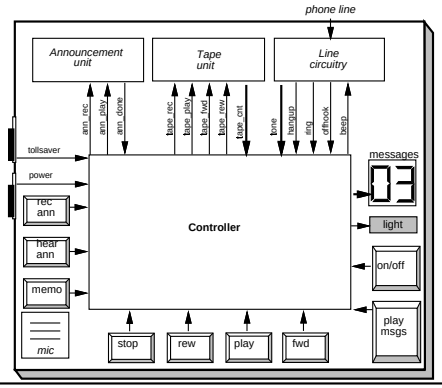
(a)

```
behavior  PlayAnnouncement  type code is
begin
   ann_play <= '1';
   wait until ann_done = '1';
   ann_play <= '0';
end;
```

(b)

```
behavior  RecordMsg  type code is
begin
   ProduceBeep(1 s);
   if  (hangup = '0') then
      tape_rec <= '1';
      wait until hangup='1'  for 100 s;
      ProduceBeep(1 s);
      num_msgs <= num_msgs + 1;
      tape_rec <= '0';
   end if;
end;
```

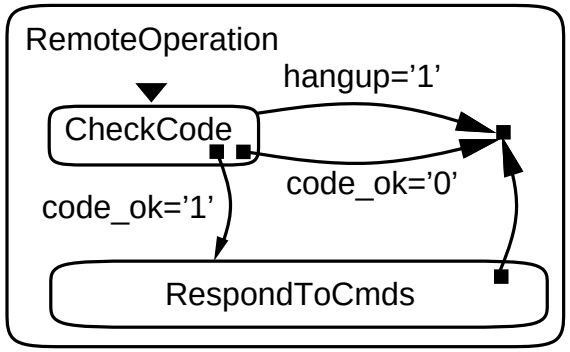
(c)



UC Irvine

The RemoteOperation behavior

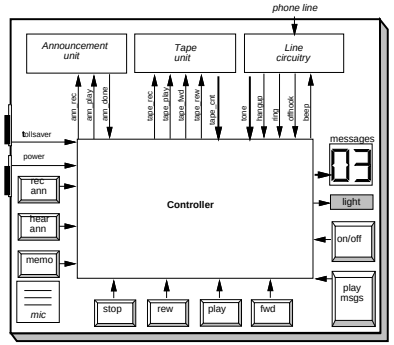
- Owner can operate machine remotely by phone
- Owner identifies himself by four button ID



(a)

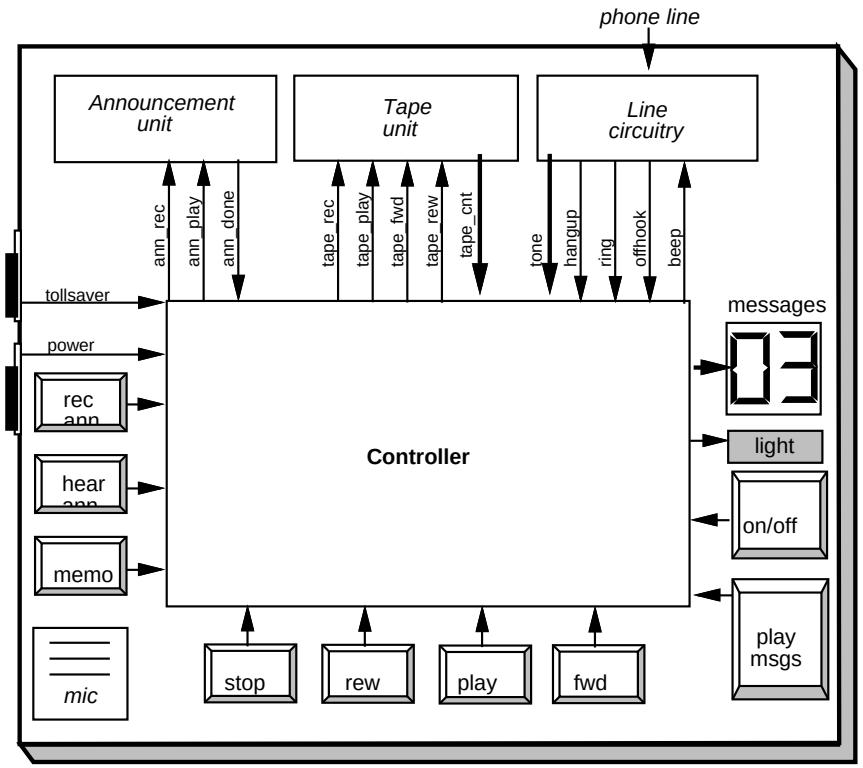
```
behavior  CheckUserCode  type code is
begin
   code_ok <= true;
   for (i in 1 to 4) loop
      wait until tone /= "1111" and tone'event;
      if (tone /= user_code(i)) then
         code_ok <= false;
      end if;
   end loop;
end;
```

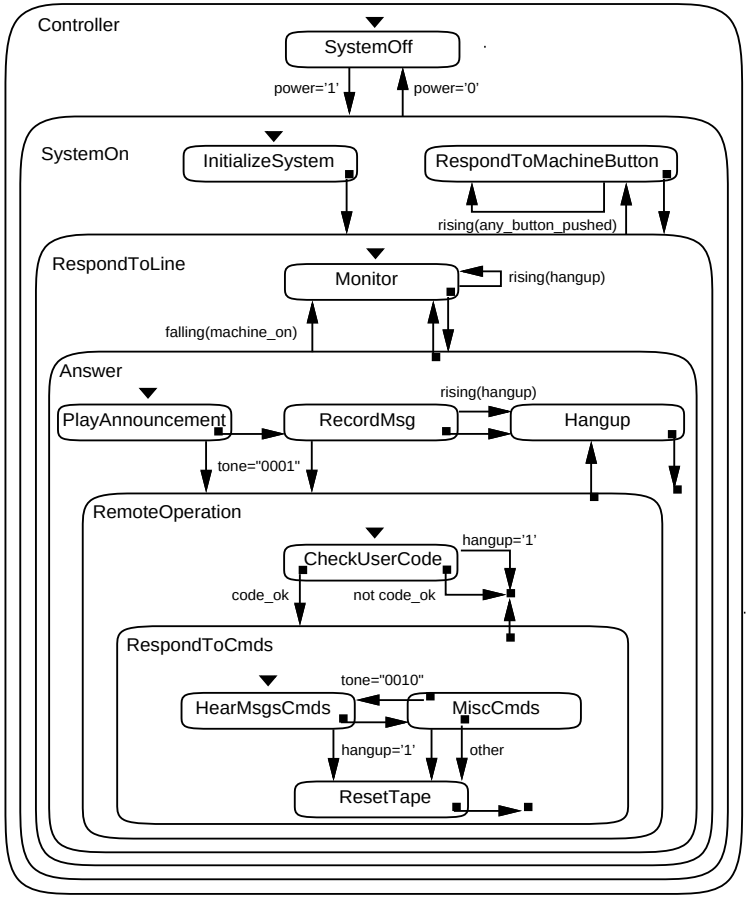
(b)



UC Irvine

The answering machine controller specification





Executable specification use

- Precision
  - Readability/precision compete in a natural language
  - Executable specification encourages precision
  - Designer asks questions, specification answers them
- Language/model match (SpecCharts/PSM):
  - Hierarchy
  - State-transitions
  - Programming constructs
  - Concurrency
  - Exceptions
  - Completion
  - Equivalence of states and programs

Specification capture experiment

	VHDL	SpecCharts
Average specification−time in minutes	40	16
Number of modelers	3	3
Number of incorrect specifications first time	2	0
Number of incorrect specifications second time	1	0

- VHDL modelers required 2.5 times longer
- Two VHDL specifications possessed control errors
- SpecCharts were effective for state-transitions and exceptions

Comparison of SpecCharts, VHDL and Statecharts

Answering machine example

		Conceptual model	SpecCharts	VHDL (hierarch.)	VHDL (flat)	Statecharts
Specification attributes	Program−states	42	42	42	32	80
	Arcs	40	40	40	152	135
	Control signals	−−	0	84	1	0
	Lines/leaf	−−	7	27	29	−−
	Lines	−−	446	1592	963	−−
	Words	−−	1733	6740	8088	−−
Shortcomings	No sequential program constructs					X
	No hierarchy			X	X	
	No exception constructs			X	X	
	No hierarchical events				X	
	No state−transition constructs			X	X	

Design quality experiment

Design attribute	Designed from English	Designed from SpecCharts
Control transistors	3130	2630
Datapath transistors	2277	2251
Total transistors	5407	4881
Total pins	38	38

- No loss in design quality with an executable language

Summary

- Executable languages encourage precision and automation
- The language should support an appropriate model
  - Makes specification easy
- Strongly parallels programming languages
  - Structured vs. assembly languages
  - Object-oriented model and C++

