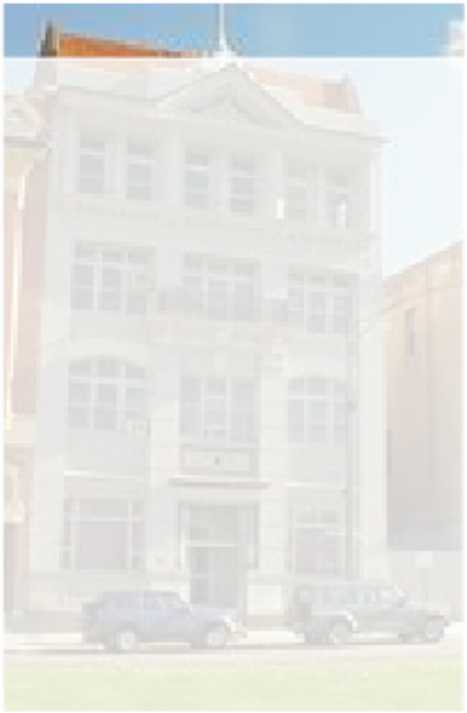
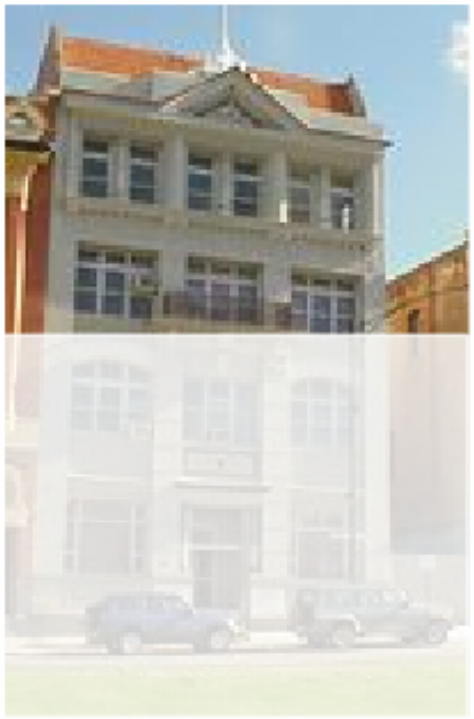
Start Simulator Stop Reset 00:00:05



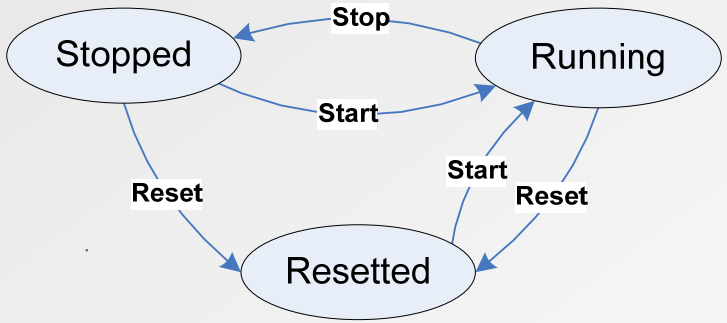
56k Modem
Connection
(60 seconds)



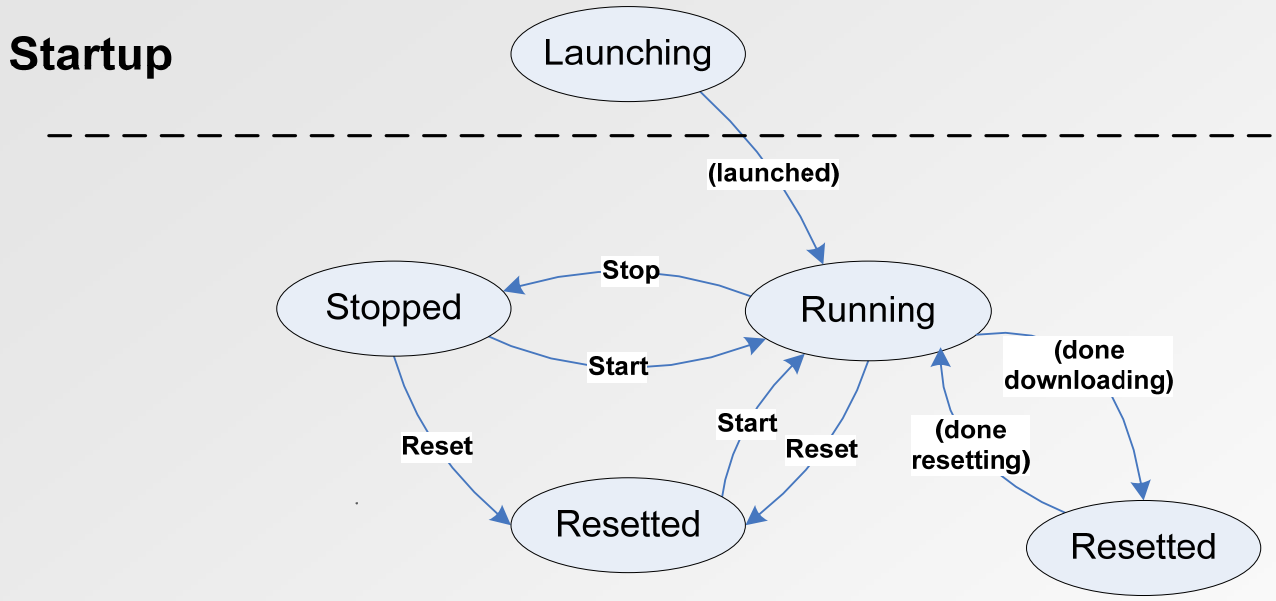
256k ADSL
(10 seconds)

1.5 M ADSL
(1 second)

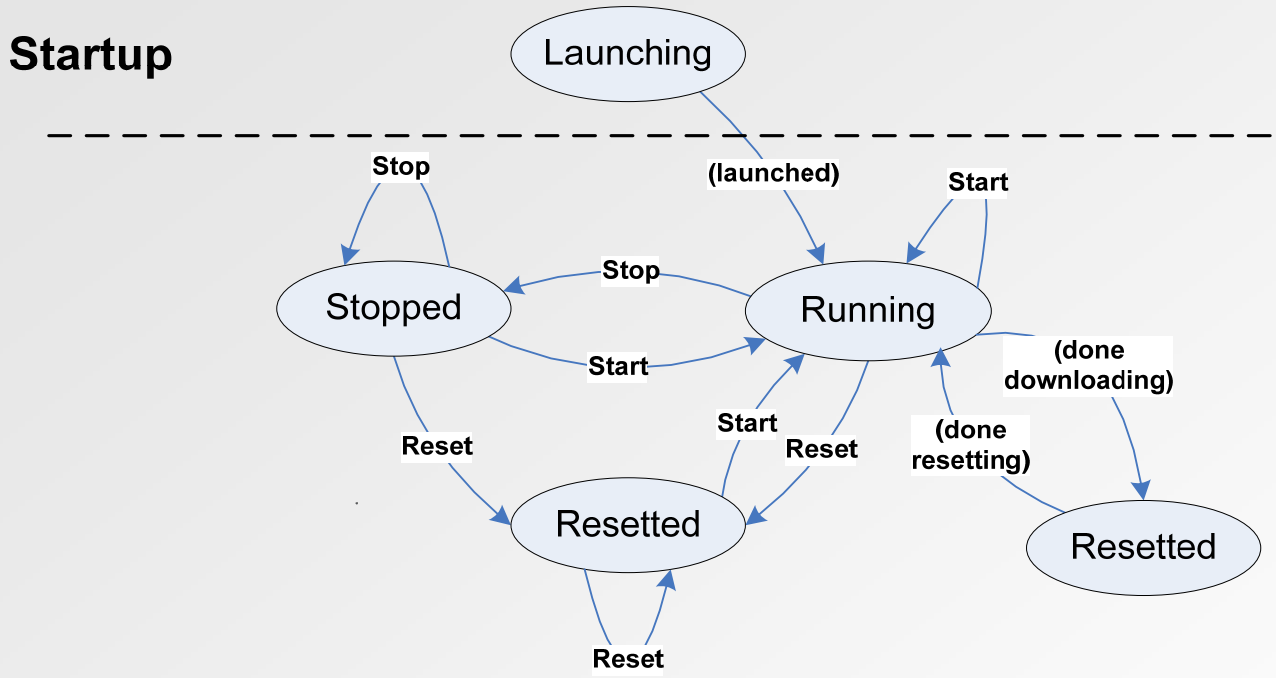
The Unbearable Lightness of Model-Based Testing



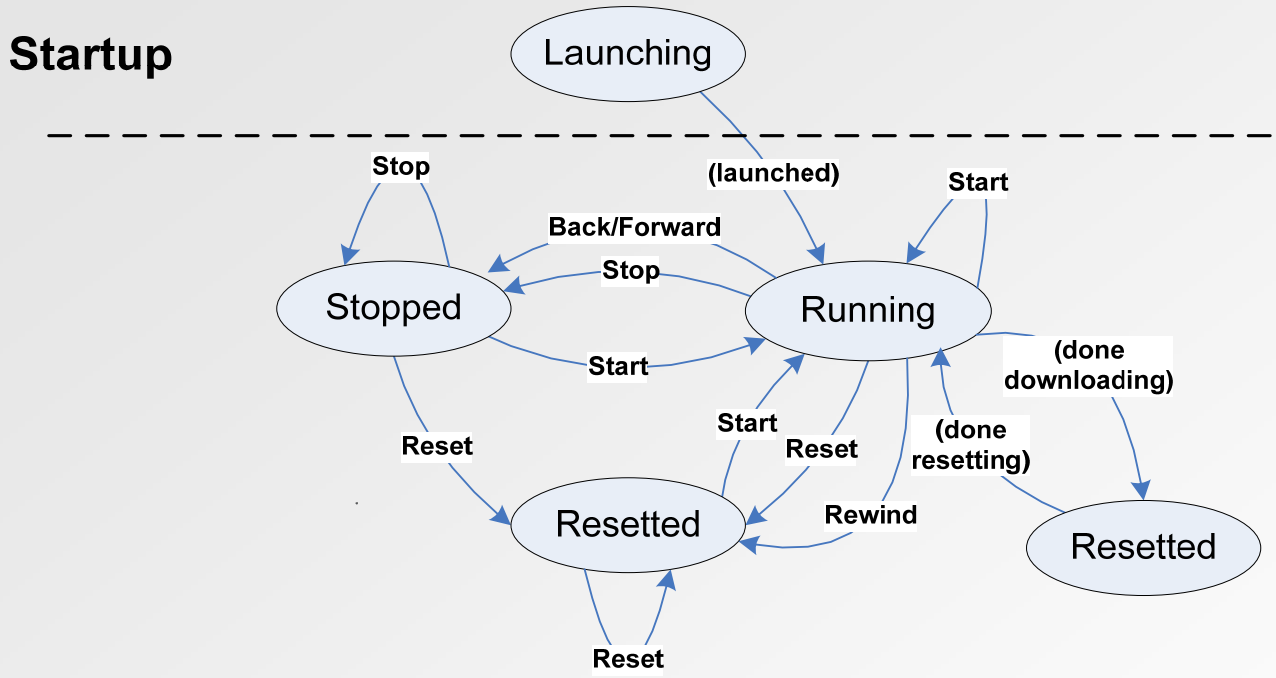
- The ADSL Flash demo is a simple program.
- This is a state model of its behavior.



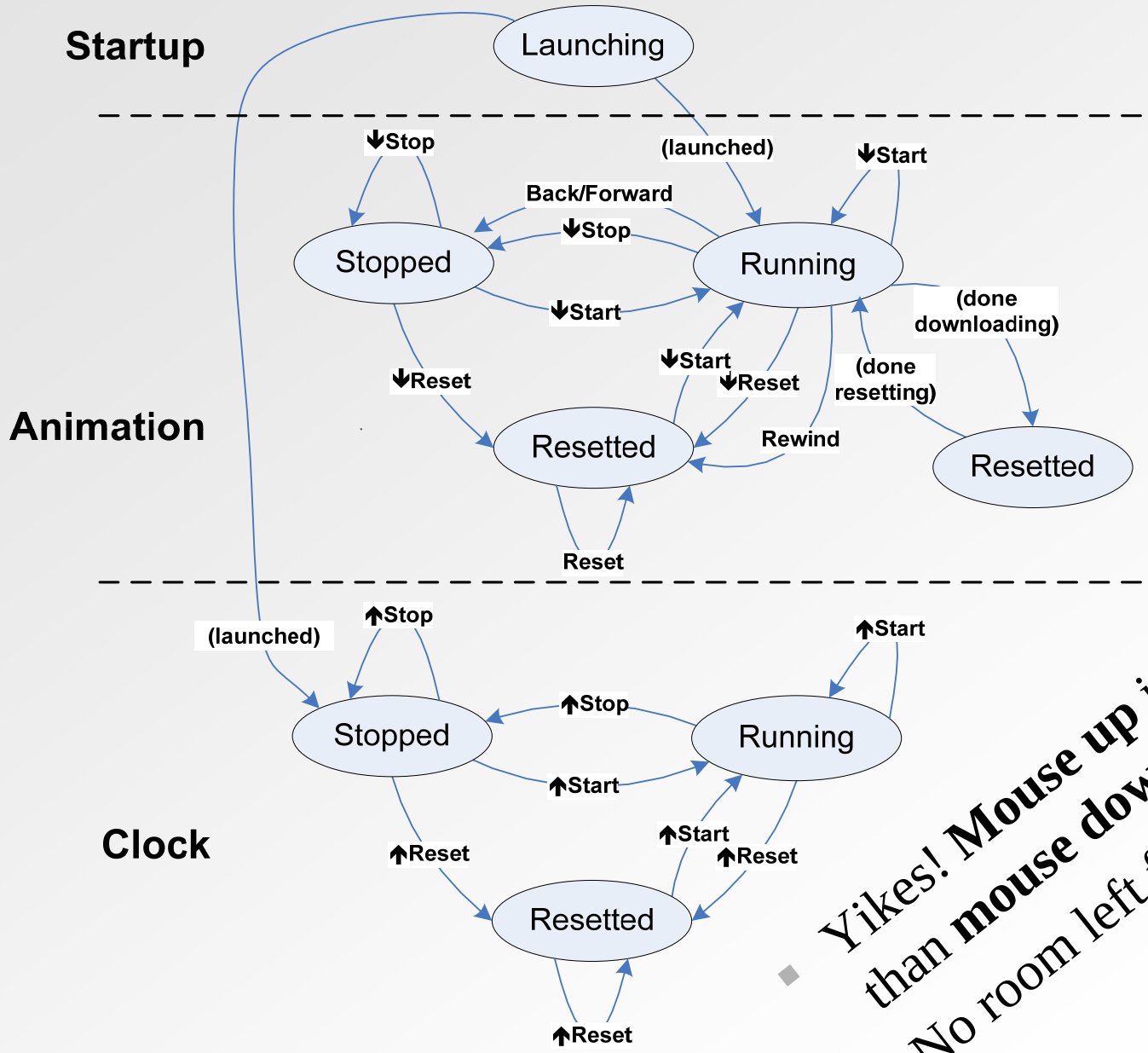
- But there are events that happen automatically.
- They may be conditional or intermittent.



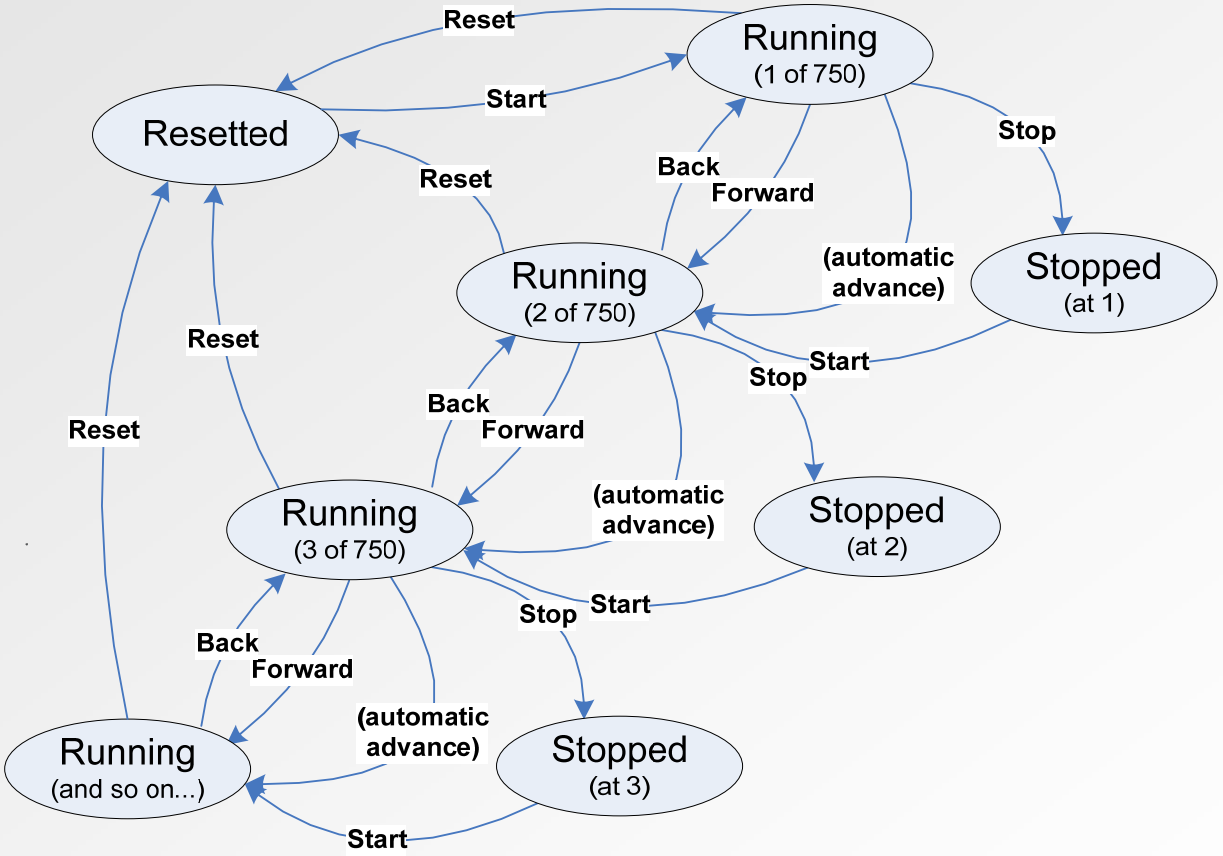
- The three major states also appear to transition back upon themselves.
- It can be very hard to know whether self-transitions are actually taking place.



- Then there are functions built into Flash...
- Not just Flash, but Windows, the browser, and who knows what libraries and third party extensions.



- Yikes! **Mouse up** is different than **mouse down.**
- No room left for bullets…



- Is it silly to drill down on states like this?
- Well, there actually are failures beginning at the 61st and 78th iteration of "Running". I did not notice them until I single stepped through each state.
- But representing this as a formal state model and explicitly generating all the tests to cover those transitions is both daunting and *probably crazy*.

Bugs I Found

- Doesn’t start clock with animation at startup.
- Clock is knocked backwards when start is pressed while running.
- Clock does not start when mouse down occurs on start button.
- Clock does not stop when mouse down occurs on stop button.
- Clock does not reset when mouse down occurs on reset button.
- Clock does not stop when “Back” is invoked from context menu.
- Time predicted doesn’t match elapsed time.
- Hours always has a leading zero (e.g. 010 for ten)
- Clock hours increment in reverse on overflow.
- Slivers progressively disappear from the middle image, during animation.

"The Law of Leaky Abstractions"

All non-trivial abstractions are leaky, to some degree.
-- Joel Spolsky (http://www.joelonsoftware.com/articles/LeakyAbstractions.html)

- If you generate tests solely from a specified model via a specified algorithm, you will miss many easy bugs.
- An "easy bug" is one you have a high probability of finding by **playing around.**

A healthy skepticism about models is essential to using models productively.

The "lightness" of playing around
brings model-based testing down to Earth!

- Skillful **"playing around"** means…
- interactively making observations, generating hypotheses, and trying inexpensive tests…
- as you work with and model the object of your study to fulfill the purposes of testing…
- *and perhaps also in preparation for fulfilling those purposes by formally generating tests from some part of the model you produce.*

Never put so much trust in any model,
that you consider testing only in accordance with that model,
and not also outside that model.