Modeling Complex Behavior Simply
or
How Much is Too Much?

Stephen J. Mellor

Project Technology, Inc.

+1 (520) 544-2881
http://www.projtech.com
To meet time-to-market for the project, we need to:

- know that we’re making steady verifiable progress
- avoid duplication and contradiction
- express results with an accessible organization
- use a simple but expressive notation
Verifiable Progress
Verifiable Progress

To make verifiable progress:

• proceed bottom up
• accumulate ‘facts’: behavioral requirements on the system
• verify facts as you proceed
Avoid Duplication
Avoid Duplication

State each fact *once* to obtain a minimal expression of a problem. Because:

- each fact is separately verifiable
- it’s faster to write something once
- it avoids contradictions
- maintenance is easier
Readable Organization
Readable Organization

Organize the models for:
• maximum exposure of information
• minimum organizational overhead

A reorganization of the model should not imply content changes.

The index should have *no* additional semantic content
The UML has a rich notation for modeling behavior.

The StateChart may be used to model:

- Use Cases
- Classes
- ......

We’ll use the StateChart to model the behavior of Classes.
UML Notation

1. **Idle**
   - lift receiver/get dial tone
   - caller hangs up/disconnect

2. **DialTone**
   - do/play dial tone
   - 15 sec.

3. **Active**
   - phone #
   - Timeout
     - do/play message
     - 15 sec

4. **Pinned**
   - callee hangs up

5. **Talking**
   - callee answers
   - caller answers-enable speech
   - busy

6. **Busy**
   - do/play busy tone
   - busy

7. **Invalid**
   - dial digit(n) [invalid]

8. **Dialing**
   - dial digit(n) [incomplete]

9. **Connecting**
   - dial digit(n) [valid]
   - connect
   - connected
   - Ringsing
     - do/play ringing tone

10. **Ringing**
    - caller answers-enable speech

11. **Timeout**
    - 15 sec

12. **Dialing**
    - dial digit(n) [valid]

13. **Connect**
    - connected

14. **Caller**
    - caller answers

15. **Caller Hangs Up**
    - disconnect

16. **Callee**
    - callee hangs up

17. **Callee Answers**
    - enable speech
Entry States and Sequential States

Sequential state
- get dial tone
- lift receiver/get dial tone
- caller hangs up/disconnect

Sequential state
- do/play dial tone

Entry state
- Active
- phone #
States and Substates

Name and identifier of superstate

sequential state

superstate

sub-state

Idle

DialTone

Active

phone #

caller hangs up/disconnect

lift receiver/get dial tone
Events, Conditions, and Transitions

- **Event**: caller hangs up/
disconnect

- **Transition**: lift receiver/
get dial tone

- **Condition**: dial digit(n)
d [invalid]

- **State**: Idle
  - **Transition**: DialTone
  - **Action**: do/play
dial tone

- **State**: Dialing
  - **Action**: dial digit(n)
  - **Condition**: do/play
dial digit(n) [invalid]

- **State**: Active
  - **Action**: phone #
Actions

- Idle
- Pinned
- Talking
- DialTone
- Dialing
- Connecting
- Ringing

Actions:
- Lift receiver/get dial tone
- Caller hangs up/disconnect
- Caller answers
- Caller answers(enable speech)
- callee answers
- callee hangs up
- Busy
- Busy tone
- Connected
- Connected busy
- Connected busy tone
- Invalid
- Dial digit(n) [valid] / connect
- Dial digit(n) [invalid]
- Dial digit(n) [incomplete]
- Do/play message
- Do/play ringing tone
- Do/play busy tone
- Do/play message
- Do/Play
- 15 sec.
- Phone #
contains all the information about a telephone and how to make calls

contains information about a physical phone

(abbreviated class diagrams)

contains information about the call only
Object Oriented Partitioning

(abbreviated object communication model)

: Telephone

contains all the behavior of a telephone and how to make calls

contains the behavior of a physical phone

contains information about the call only

: Telephone

create

delete

: Call

create

delete
Original Model

- **Idle**: caller hangs up/disconnect
- **DialTone**: lift receiver/get dial tone
- **Pinned**: do/play dial tone
- **Talking**: caller answers
- **Ringing**: caller answers/enable speech
- **Connecting**: connected
- **Timeout**: 15 sec.
- **Invalid**: dial digit(n) [invalid]
- **Busy**: busy
- **Dialing**: dial digit(n) [valid]
- **Active**: phone #

Diagrams show transitions between states with annotations for actions and conditions.
No Semantic Content in Index
Avoid Hierarchical States
But WAIT! This is WORSE!

So.....build still *more* objects
Commununicate Synchronously

On Hook
- entry/
- Off Hook
- Phone
- delete
- disconnect

Off Hook
- entry/
- signal
- Gone Off Hook

synchronous communication

DialTone
- do/play
- dial tone

15 sec.

Invalid
- do/play
- message

15 sec

Dialing
- dial digit(n)

Making Call
- Entry/Signal
- Create Call

15 sec

dial digit(n) [valid] /
connect

dial digit(n) [incomplete]

Gone Offhook
Behavior Normalization

Partition so that each instance has the:

• same characteristics and conforms to the set of rules and policies

each instance has the same behavior

| Physical Telephone | can be 0..1 | Off-Hook Phone | initiates 0..1 | Call |
Partition Further When....

......a set of related states all make the same transition.

Whatever is happening with an Off-Hook Phone, when you hang up, it goes away.
Partition Further When....

...a relationship exists only in certain states.

The call can only exist when the telephone is off-hook.
Partition Further When...

....a set of related states are concurrent

Call Waiting means that a phone has two calls at once.
To Simplify Complex Behavior....

...use only a subset of the notation.

- entry actions
- transition actions
- exit actions
Entry Actions Only

On Hook
- entry/delete
- Off Hook Phone; disconnect

Off Hook
- lift receiver
- get dial tone
- caller hangs up

Off Hook
- entry/signal
- Gone Off Hook

Get Dial Tone

On Hook
- entry/ delete Off Hook Phone disconnect

lift receiver
- caller hangs up
Summary

To model complex behavior simply:

• work bottom-up from known facts
• express behavioral requirements once
• avoid semantic content in the index
• simplify the notation

This will speed time-to-market for your project.