PART FIVE:

SOME VERIFICATION CHALLENGES

a DFC case study with several verification problems, all concerning signaling properties and event-based feature interactions
**SIGNALING INTERACTIONS**

*transparency*: a box behaves transparently when its functions are not needed

*autonomy*: each box has the power it needs to perform its functions independently

---

**Target Zone**

- **RVM**
  - **IVR dialogue with caller**: caller chooses whether to interrupt targeted subscriber
  - if no, QT sends *unavail* upstream
  - if yes, QT continues the chain and becomes transparent

- **QT**
  -IVR dialogue with caller; caller records voice mail in subscriber's mailbox

- **RVM**
  - if whole attempt times out, send *unavail* upstream
  - if single branch succeeds, send *avail* upstream

- **PR**

- **Context Independence**: by its behavior and its place in the precedence order, a feature box can trigger, delay, or cancel other features without knowing whether they are present or what they are

---

**IVR dialogue with caller**

- caller chooses whether to interrupt targeted subscriber

---

**Absorb Failures of Single Branches**

- QT sends *unavail* upstream if single branch succeeds, send *avail* upstream

---

**QT**

- IVR dialogue with caller; caller chooses whether to interrupt targeted subscriber

---

**PR**

- QT sends *unavail* upstream if whole attempt times out, send *unavail* upstream
BOXTALK EXAMPLE: CALL FORWARDING ON BUSY

this is a Boxtalk program, in graphical syntax

Boxtalk implementation sets up and tears down calls, so the programmer only works directly with status signals

each active call must be referred to by a call variable; each responsive state lists the variables of all active calls

continuation without address translation

rcv(i) / ctu(i,o)

uncertain: (i,o), (i[v],o[v])

uncertainty is resolved by the first outcome signal

o?avail, o?unknown

transient: (i,o), (i[v],o[v])

continuation with address translation

o?unavail / end(o); ctu(i,o){trg=fwd_addr}
A feature box should be INPUT-ENABLED, i.e., guaranteed to read every input signal in a prompt fashion.

**BOXTALK EXAMPLE: CALL LOGGER**

In a transient state, the box is not responsive to inputs. In a stable state, a response to every received signal is defined:

- If there is an explicit transition, execute it.
- Else if the signal is a teardown, tear down all calls and quit.
- Else if the source of the signal is linked, send signal to all linked calls.
- Else discard the signal.

This box can be used in a source or target zone.
A CASE STUDY

HISTORY

- these features are the relevant part of a feature set that was deployed by AT&T in a consumer trial of VoIP October 2003 to March 2004
- most of them are now in CallVantage, which is AT&T’s VoIP service
- when the feature set was designed, the feature interactions were analyzed manually and heuristically

BOUNDARIES

- the only devices are "black phones"
- no source feature boxes
- no bound feature boxes
- a feature box can only use the new method to place a call to a resource
- a feature box cannot use the rev method

WHAT CAN HAPPEN THAT IS INTERESTING?

- a target region can contain several target zones
- a usage can reach an interface box, or stop at some feature box
- a usage can fork
- a usage can have multiple, sequential extensions downstream of a feature box
BOUND BOX: BLACK PHONE INTERFACE (SIMPLIFIED)

- **silent**: c
- **dialing**: (c[v], phone)
- **ringing**: c, (c[v], phone)
- **dialled / new(c) { dld=dialed_string }**
- **busytone**: c
- **waiting(c[v])**
- **ringback**: c
- **talking**: c, (c[v], phone)
- **offhook / callavail**
- **c?unavail**
- **accepted(c[v])**
- **onhook**
- **disconnected**

- **rcv(c)**
- **from (almost) any state**
FREE BOX: PARALLEL FIND ME

box generates unavail under many different circumstances

database_query

rcv( i )

no_loc_exists / i!unavail

exists_1_loc /
ctu( i, o1){ trg=loc1 }; t!tset{ duration="40" }

onering:
( i, o1 )
( i[v], o1[v] )

o1?unknown, o1?unavail /
end( o1); o1,o2 = o2,-
gone( o1 ) / o1,o2 = o2,-
o1?avail / i!avail; end( o2 )
o2?unknown, o2?unavail / end( o2 )
gone( o2 )
o2?avail / i!avail; end( o2); o1,o2 = o2,-

tworings:
i, o1, o2
( i[v], waiting )
exist_2_locs /
ctu( i, o1){ trg=loc1 }
ctu( i, o2){ trg=loc2 }
t!tset{ duration="40" }
t?tout / i!unavail

gone( o1 ) / o1,o2 = o2,-
o1?unknown, o1?unavail / end( o1)

end( o1); o1,o2 = o2,-

o1?unknown, o1?unavail, t?tout / i!unavail

o1?unknown, o1?unavail, t?tout / i!unavail
FREE BOX: QUIET TIME

dialogue says that the subscriber wishes not to be disturbed, prompts caller to leave a message (choice "quit") or interrupt the subscriber (choice "ctu")
FREE BOX: RECEIVE VOICE MAIL

transparent:
- $(i, o)$
- $(i[v], o[v])$

dialogue:
- $(i, r)$
- $(i[v], r[v])$

box converts unavail from downstream to avail going upstream

rcv$(i) /$ ctu$(i, o)$

o?unavail /
i!avail; end$(o)$;

new$(r)$
- dld="voicexml_server"
- script="voice_mail"

dialogue offers caller the opportunity to record voice mail

r?response{ choice="end" }
the primary purpose is to prevent this:

cellphone is unavailable; cellphone voice mail answers the call, preempting the possibility that a person could answer the other phone

Answer Confirm goes here; it will convert success to failure if no digit is entered

trying:
( i, o )

the dialogue prompts the callee to enter a digit confirming presence and identity

confirming:
i, ( o, r )
(o[v], r[v] )

transparent:
( i, o )
( i[v], o[v] )

rcv( i ) / ctu( i, o )

o?avail, accepted( o[v] ) /
new( r )
{dld="voicexml_server", script="answer_confirm"}
FREE BOX: SEQUENTIAL FIND ME

firsttry:
( i, o )
( i[v], o[v] )

gone( o )

o?unknown,
o?unavail,
t?tout /
end( o )

! / ctu( i, o ){trg=head_loc_list }; dehead_loc_list;
t!tset{ duration="40" }

transparent:
( i, o )
( i[v], o[v] )
o?avail / i!avail

loc_list_empty / i!unavail

nexttry:
( i, r ), o
( i[v]<r[v] )
o?unknown,
o?unavail,
t?tout /
end( o )
gone( o )

cvu( i, o )
{trg=head_loc_list }; dehead_loc_list;
t!tset{ duration="40" }
	new(r)
{ dld="voicexml_server",
script="sequ_findme" }
o?unknown,
o?unavail,
t?tout /
end( o )
gone( o )

rcv( i )

database
query

o?unknown,
o?unavail,
end( o )
o?unknown,
o?unavail,
t?tout /
end( o )

no_loc_exists / i!unavail

o?unknown,
o?unavail,
t?tout /
end( o )

loc_list_empty / i!unavai

o?avail / i!avail; end( r )
o?avail / i!avail

end( o )
VERIFICATION CHALLENGES

A COMPLETE SPECIFICATION OF A FEATURE SET BASED ON THESE PROGRAMS MUST ALSO INCLUDE:

- a precedence partial order on the feature box types AC, PFM, QT, RVM, SFM
- constraints on which addresses can subscribe to which features
- constraints on the addresses used by Find Me features

WE CONSIDER THE INTERACTIONS AMONG THESE FEATURES THAT ARE GOVERNED BY GENERAL PRINCIPLES, NOT PERSONAL CHOICE

CHALLENGES

- Within the boundaries of this study, what are the correctness criteria for feature sets?
- Complete the specification of a feature set based on these programs, and prove that it satisfies the correctness criteria.
- Devise constraints on box behavior, precedence, subscriptions, and operational data that guarantee the correctness criteria.
- Prove that the design constraints guarantee correctness.
- Show that the design constraints allow reasonable features to do their jobs.

I DO NOT KNOW THE ANSWERS!