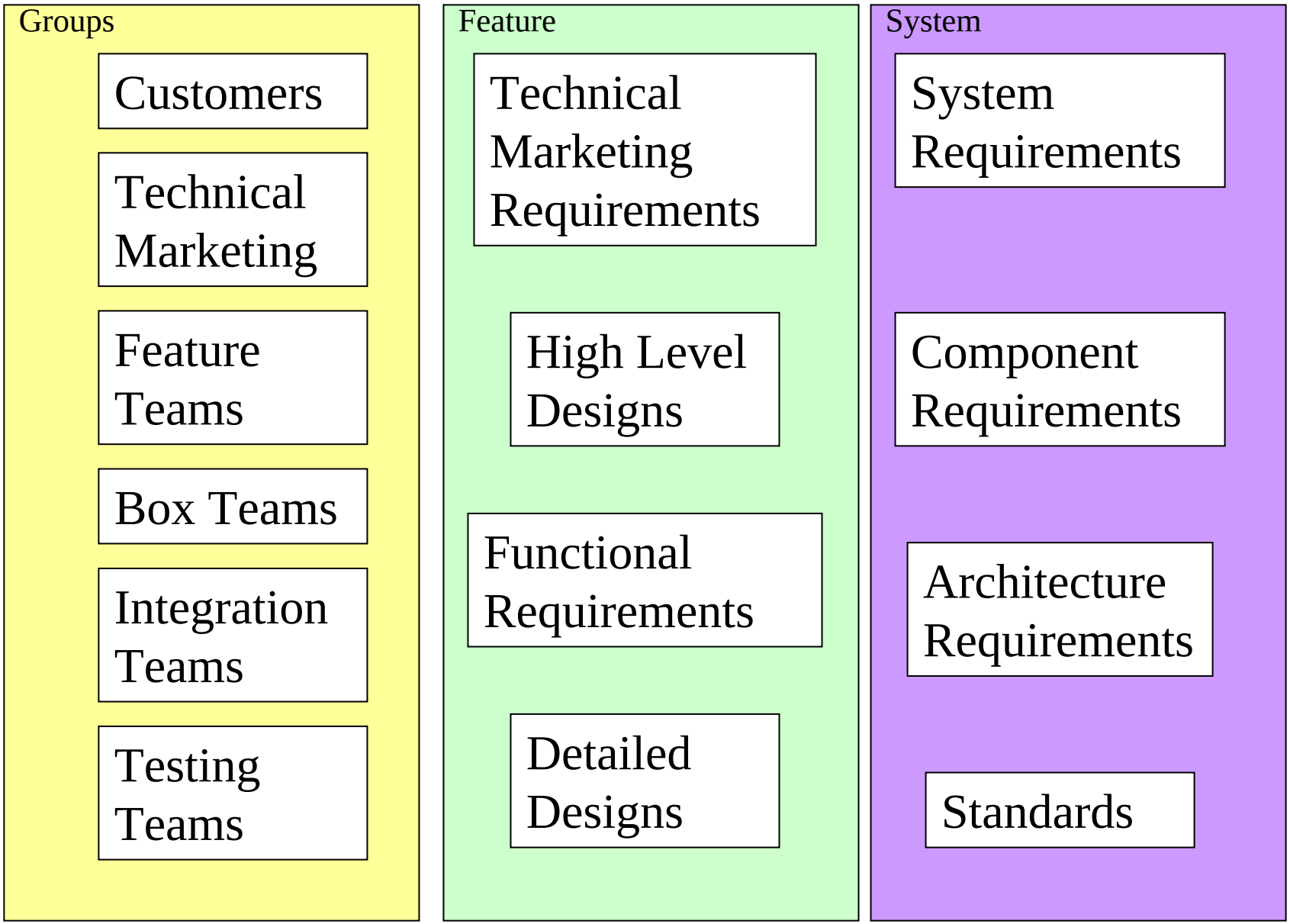
MOTOROLA LABS

UniS

Scenario Synthesis from Imprecise Requirements

Bill Mitchell, Robert Thomson, Paul Bristow

Enterprise Development Process



Telecoms Example

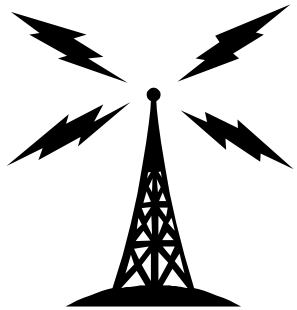
- Network provider deploying 3G.
- Placing order for handsets.
- One of the many features included will be access to network Java game repository.

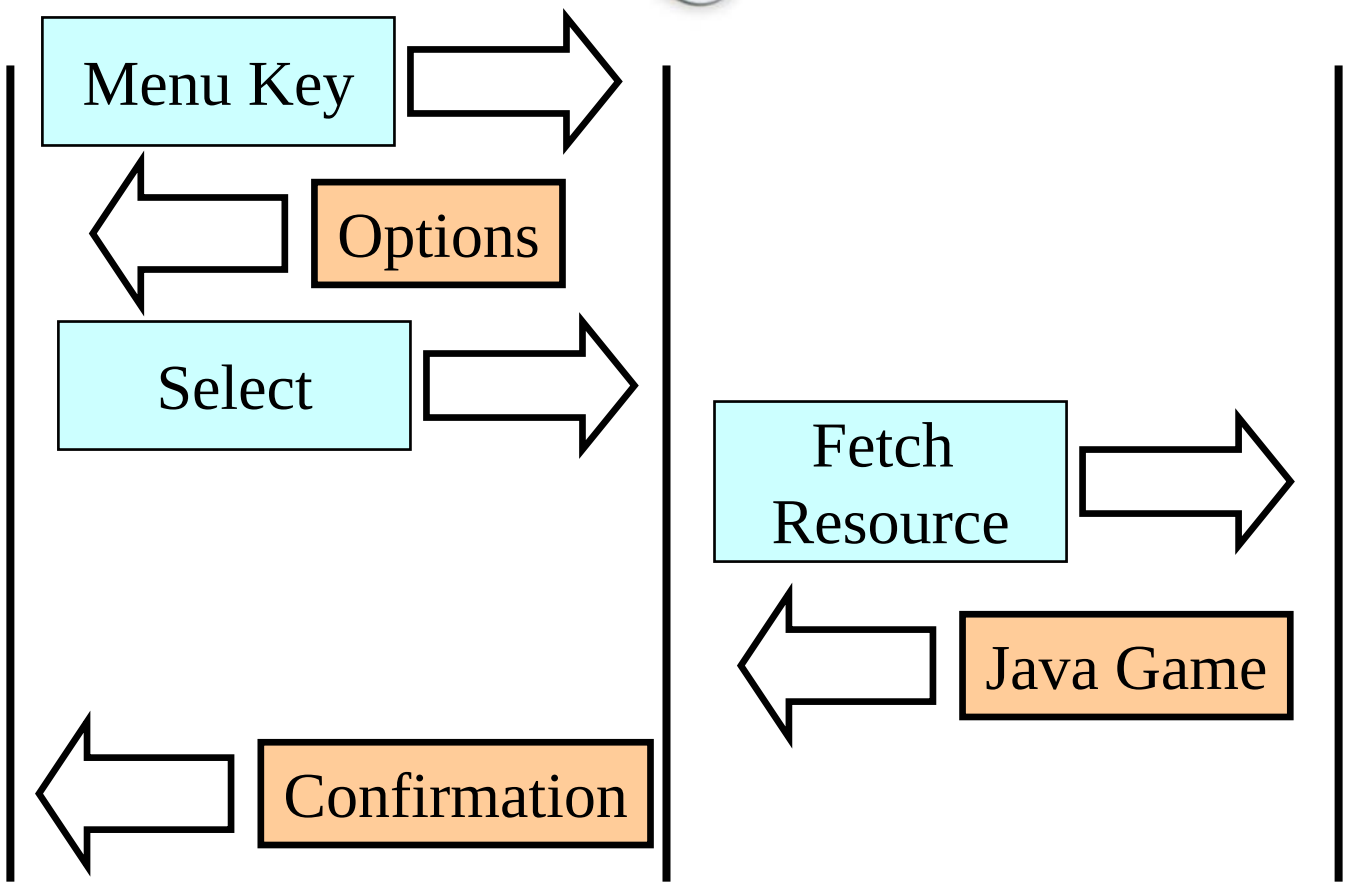
Initial Customer Requirements

User

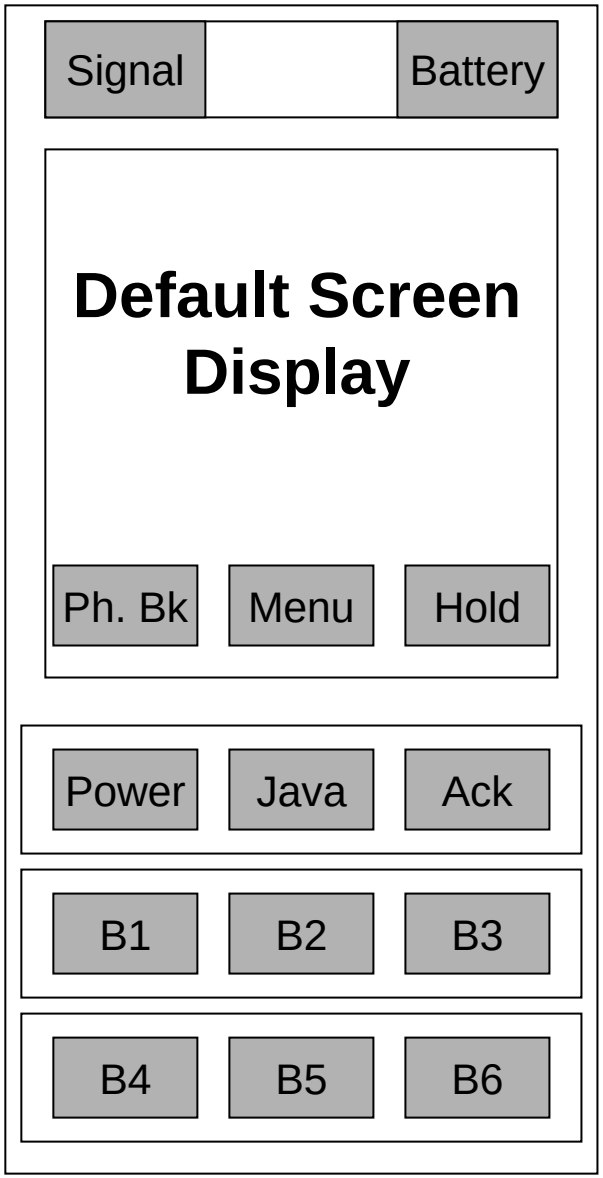
Handset

Network





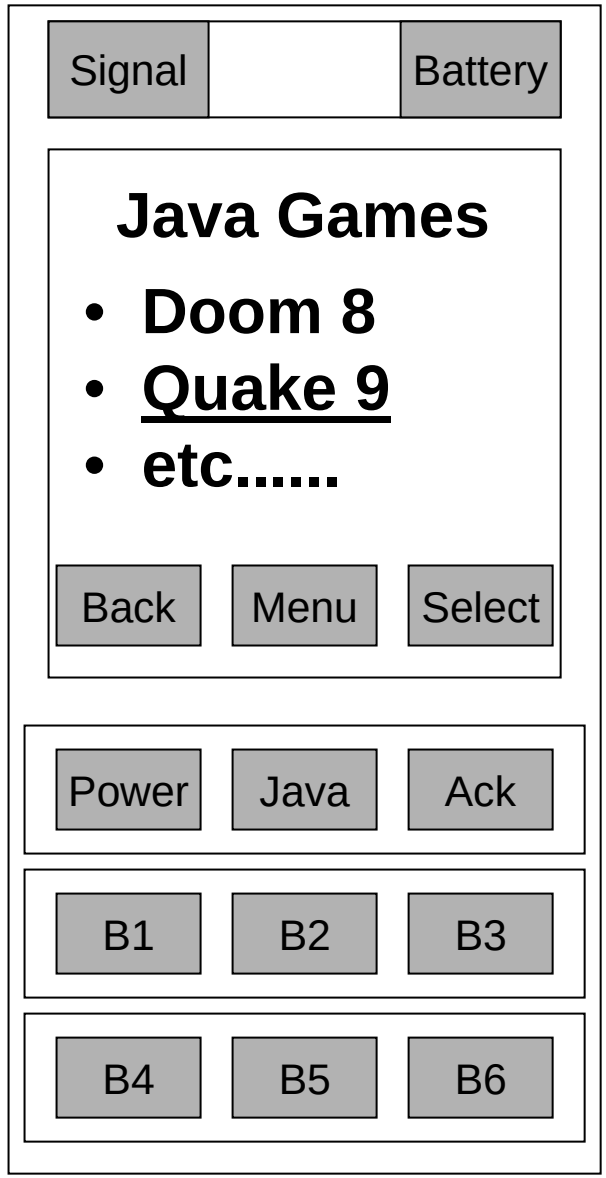
Technical Marketing Scenarios



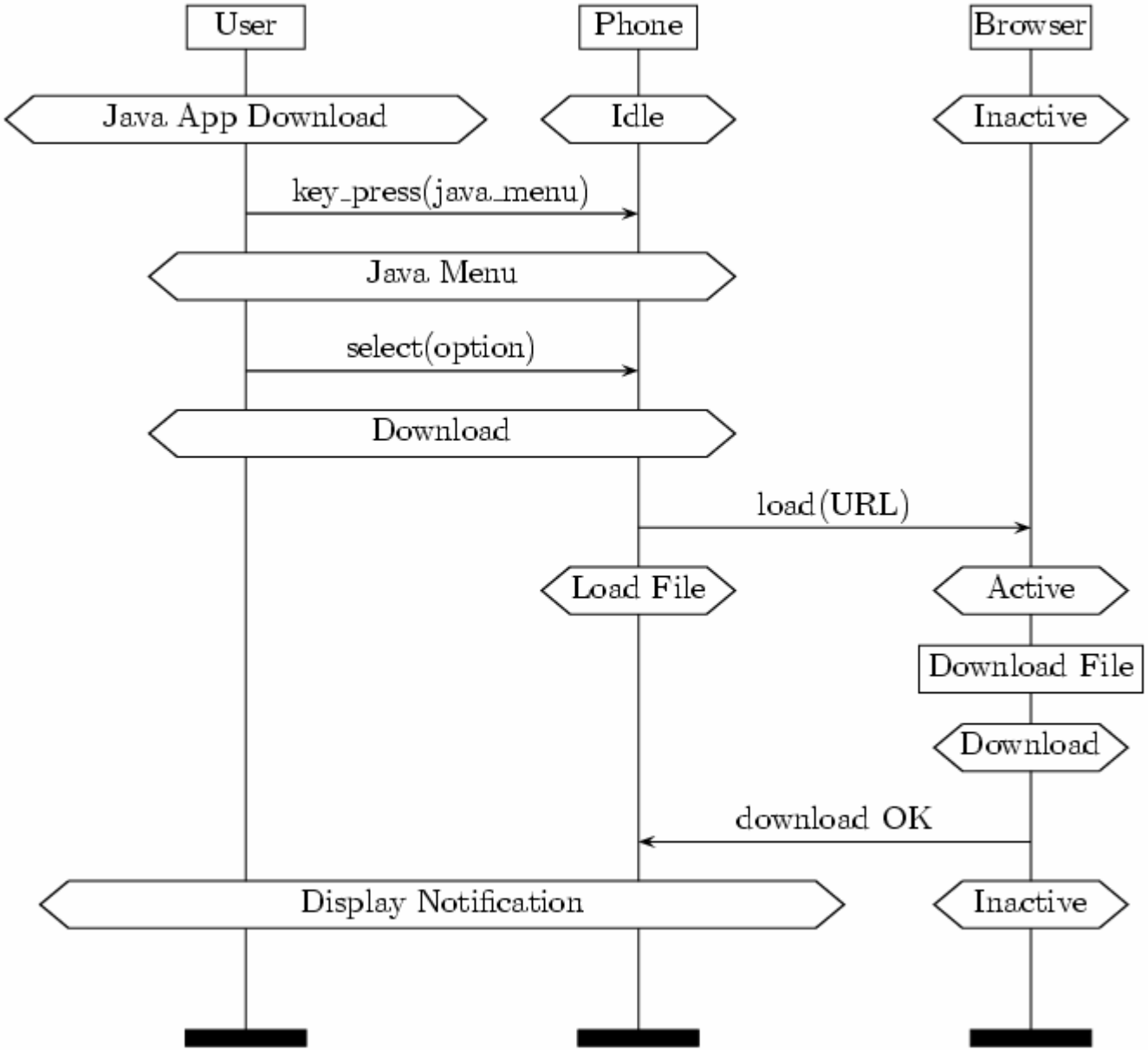
Customer Scenario broken down into sequence of atomic events, which change interface functionality.

Each event modifies functionality and UI configuration

Java Key Press



Functional Requirements



Technical Marketing Scenarios

Normative scenarios are very focused on isolated behaviour of feature in these requirements:

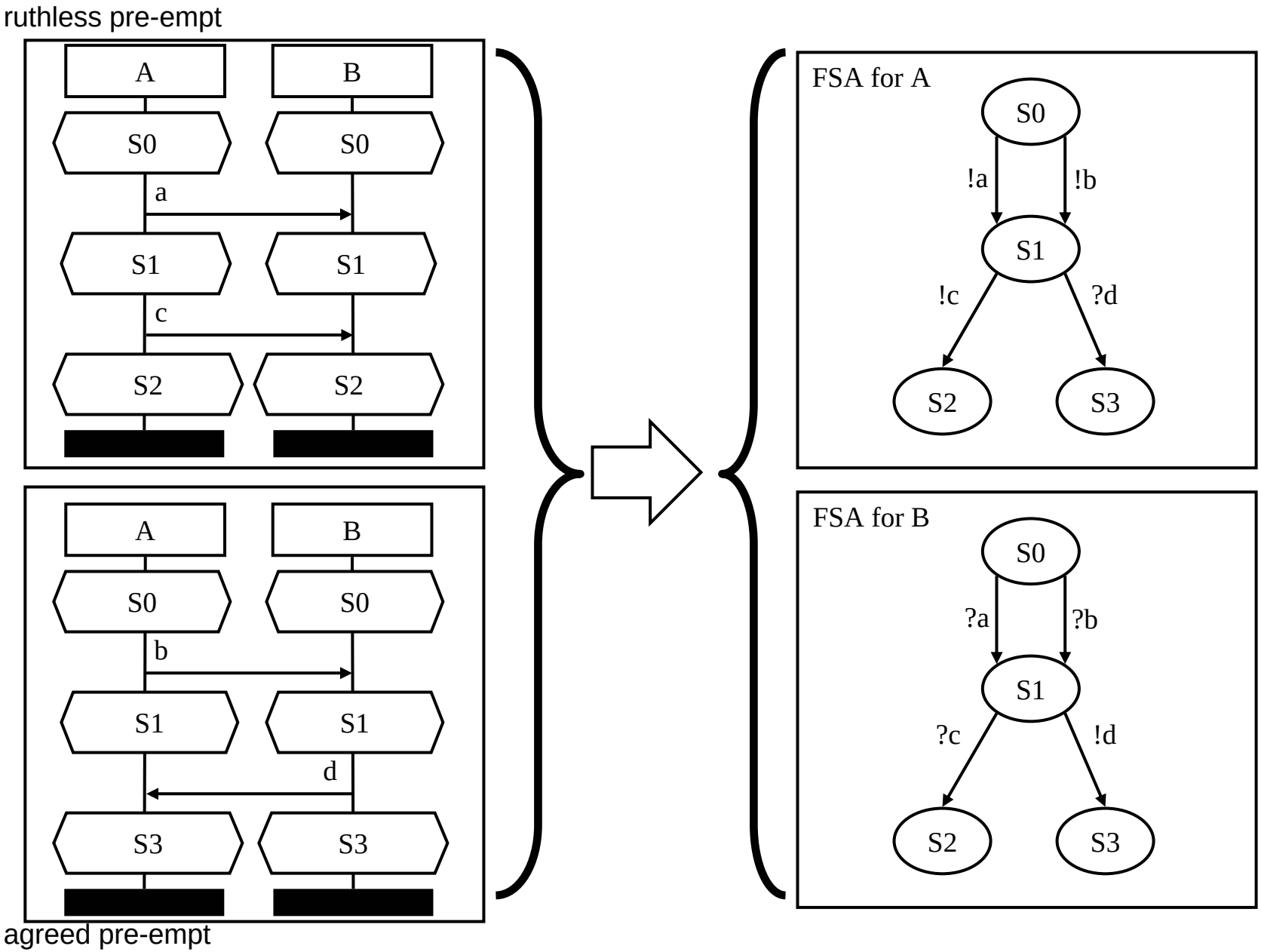
- What if voice or data call received during download?
- If memory is expandable (as with some PIM-phone hybrids) how should the mem-full error be handled if the user could add extra memory with, say, a USB flash memory stick?
- What if during the download the network service provider tries to update the phone configuration via the air interface for enhanced game play?

Need to synthesise model of system from all MSC requirements scenarios for simulation and analysis.

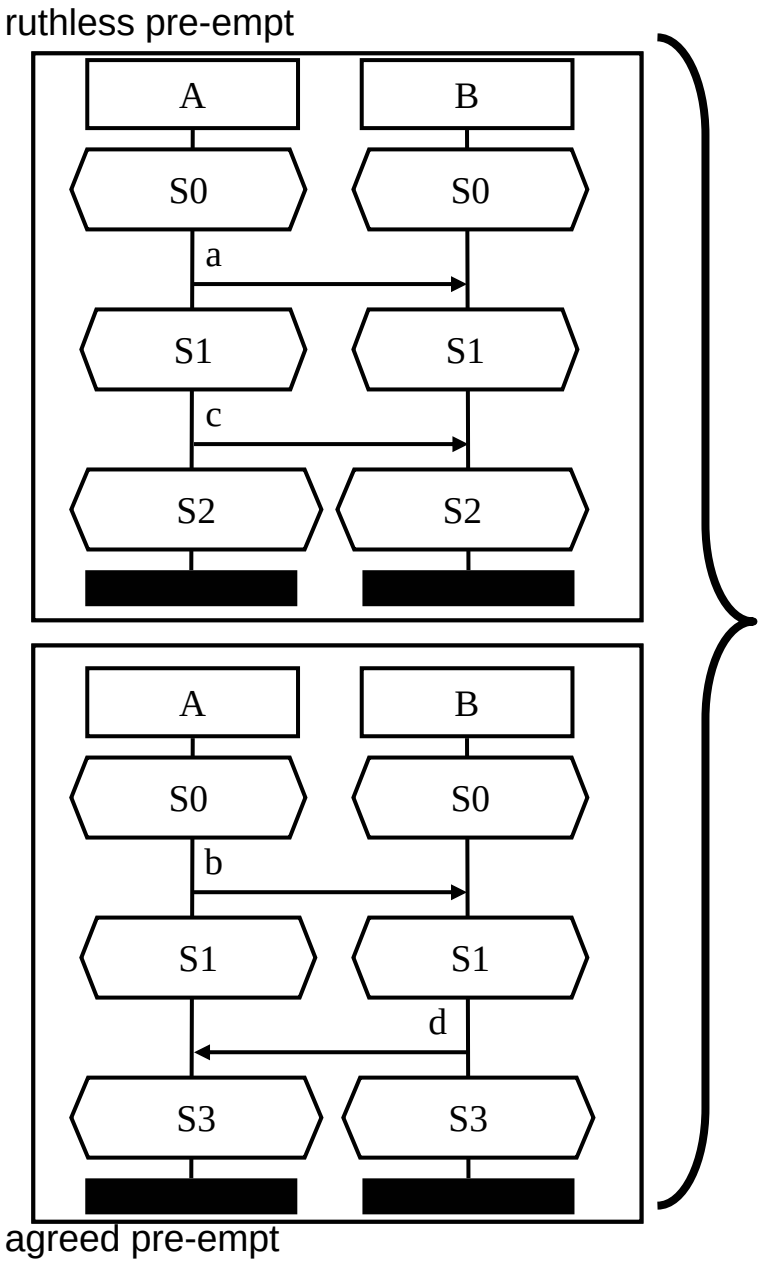
Problem:

- Practitioners use states imprecisely
- Different engineering groups define scenarios differently
- Legacy requirements

Deadlock example from TETRA PPT

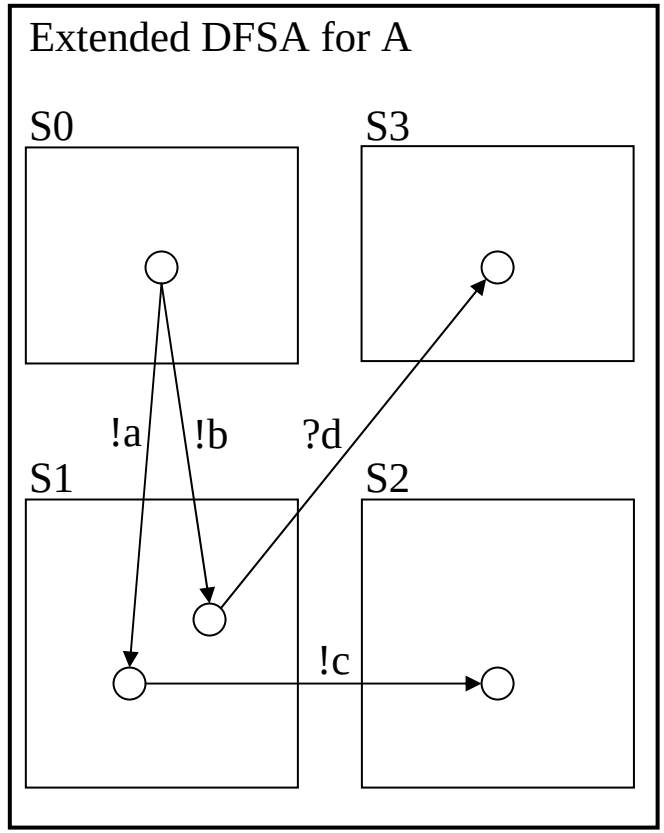


Example Deadlock Avoided



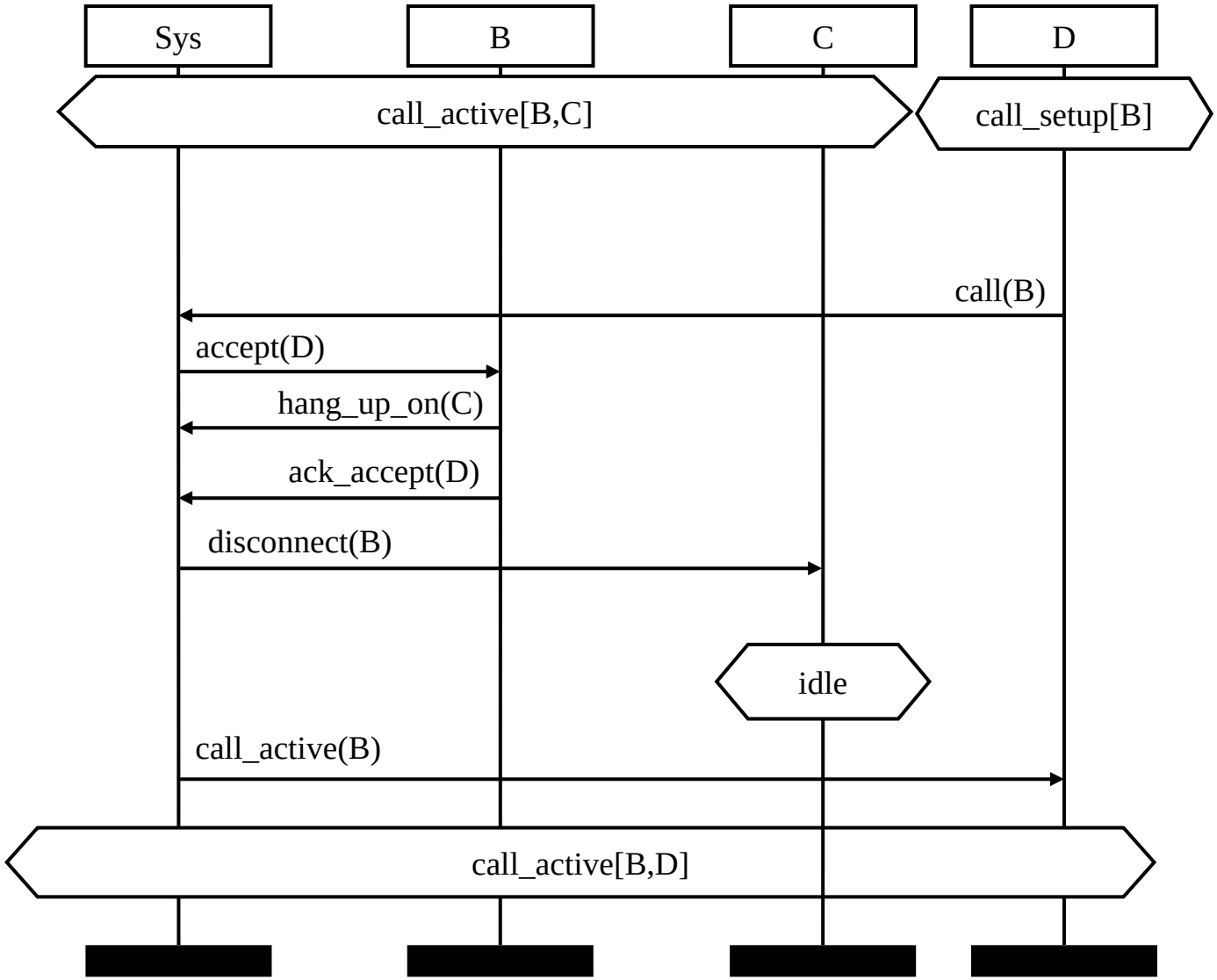
Composite States

- Anonymous internal states
- Multiple entry/exit states

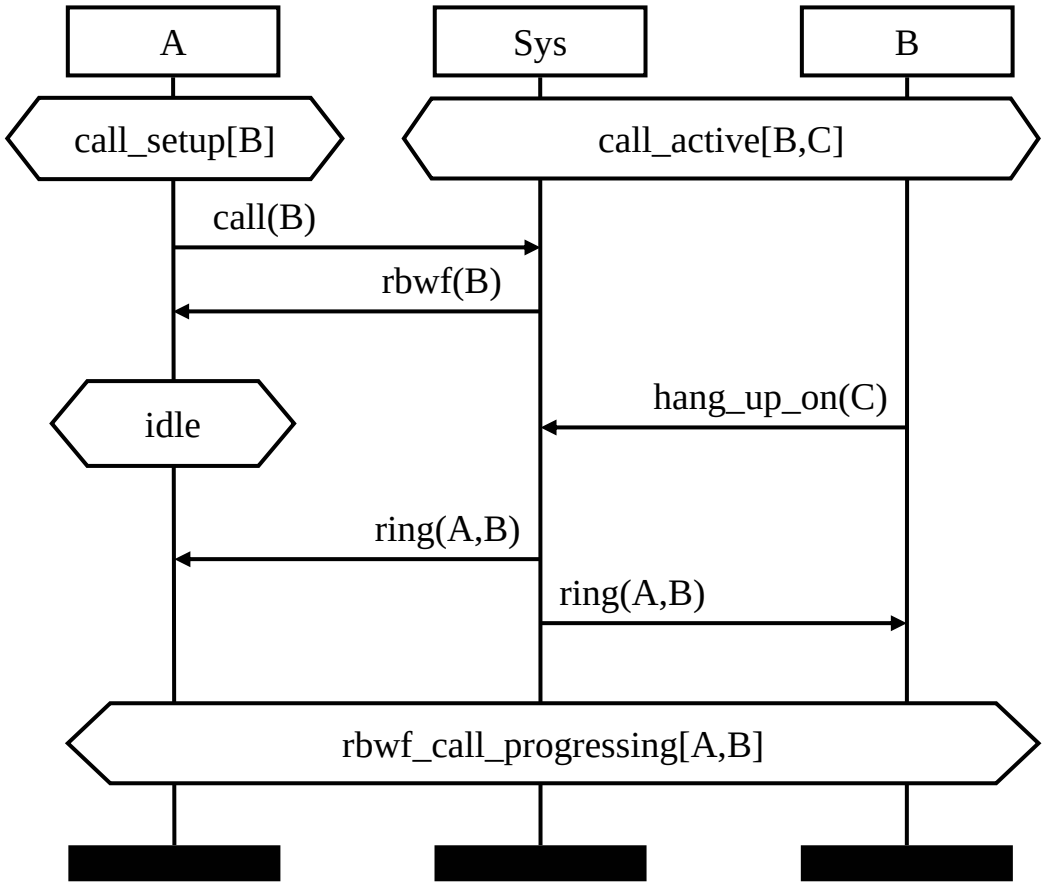


Too Weak to ever give any interactions!

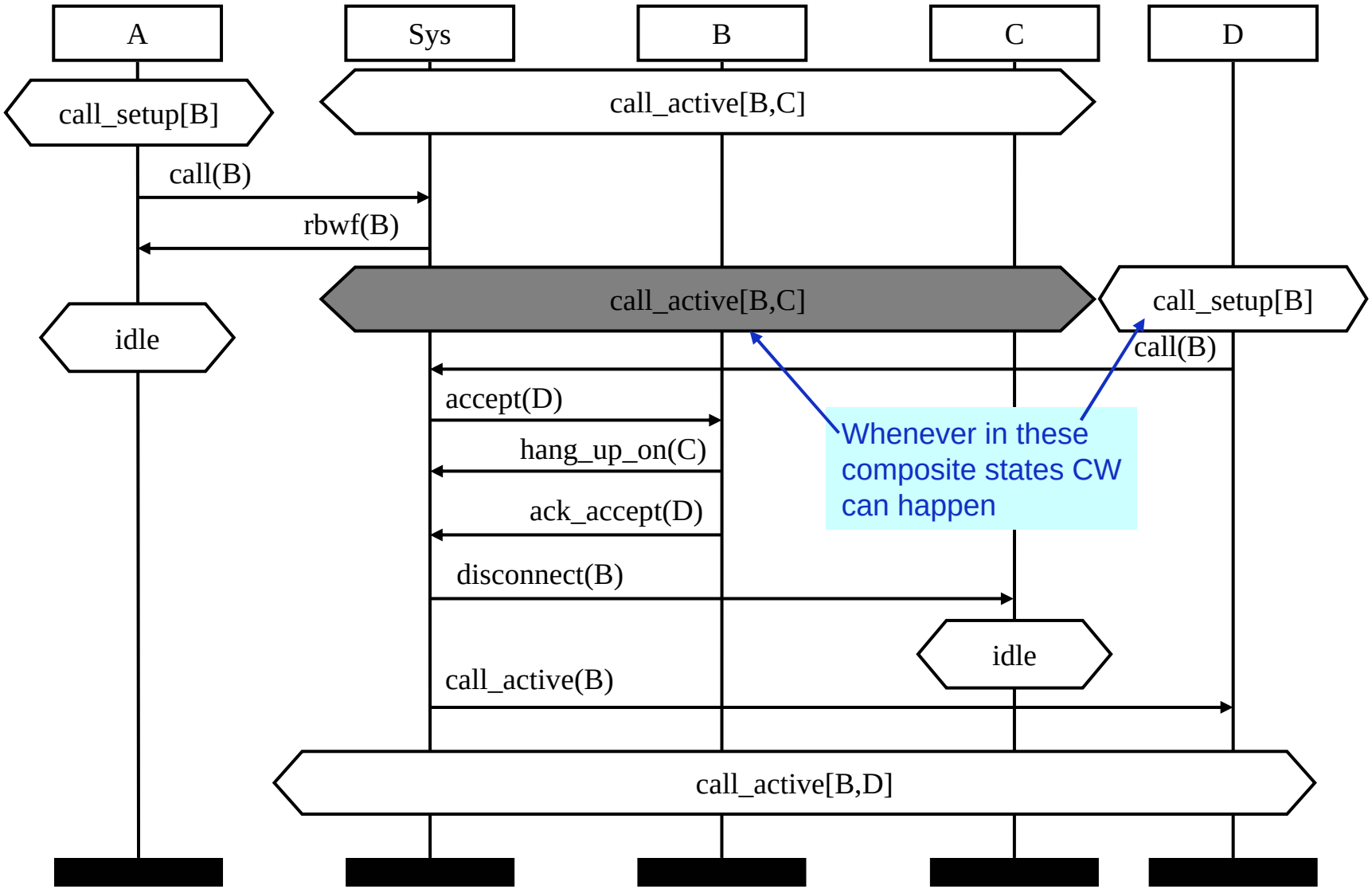
Example, Call Waiting from paper in FIW 2000



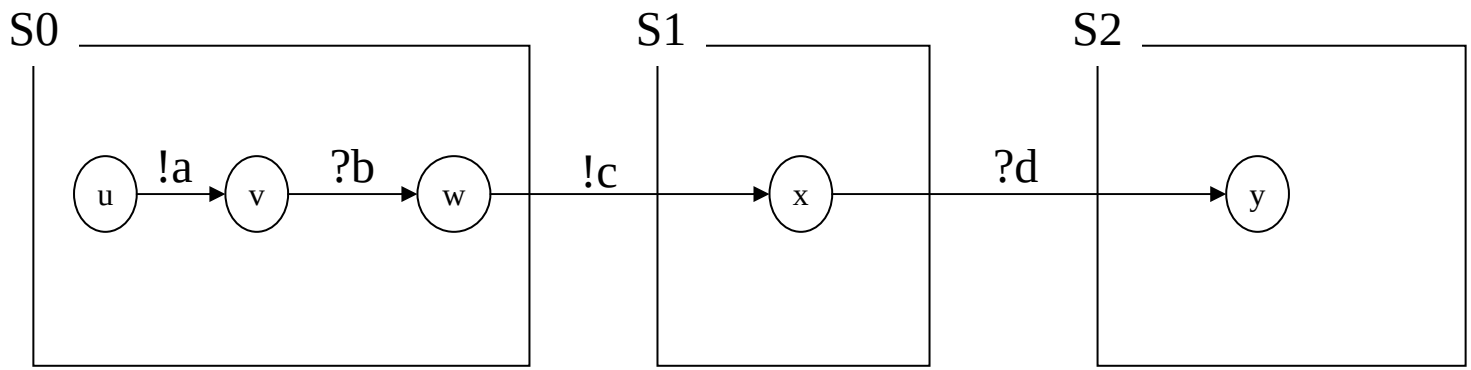
Example, RBWF, from paper in FIW 2000



Example, FI from paper in FIW 2000



Trace semantics for states



State x is (In, Out), where In and Out are sets of traces.

For every trace t1 of In there is a path

$$u \xrightarrow{t1} x$$

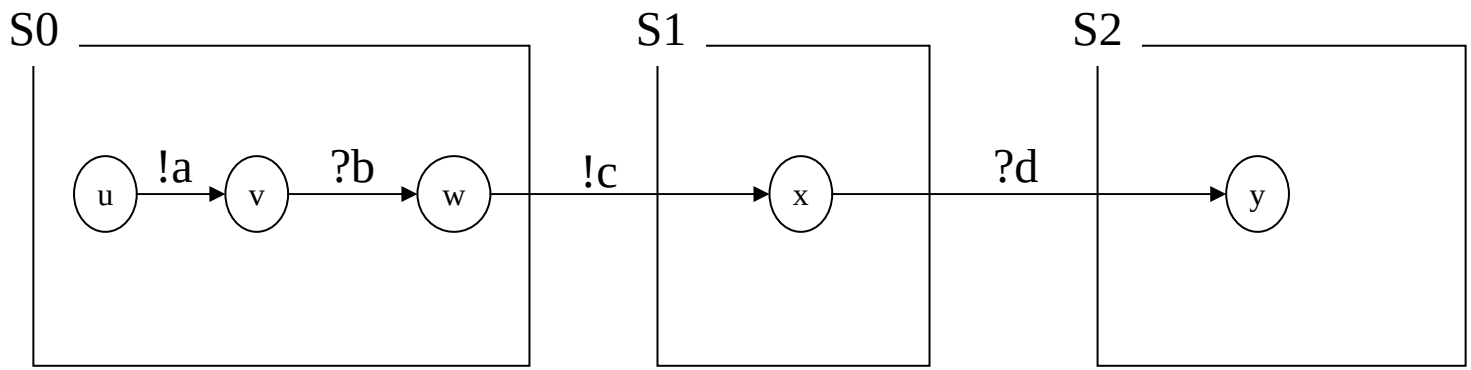
some initial state u

For every trace t2 of Out there is a path

$$x \xrightarrow{t2} y$$

some accepting state y

Deterministic trace semantics



For any t1 of In <u>if</u> there is a path

$$u \xrightarrow{t1} x$$

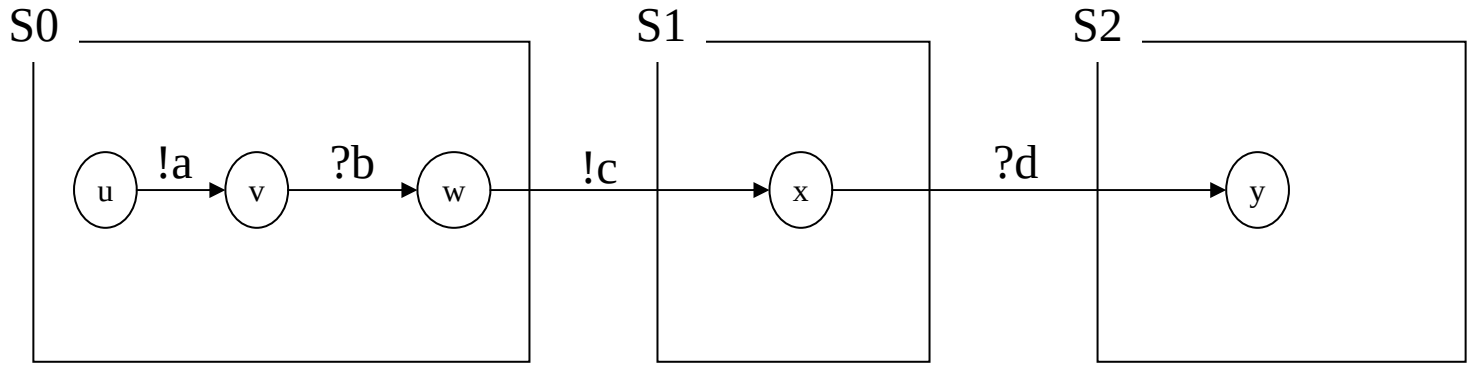
for some initial state u

<u>then</u> for every trace t2 of Out there is a path

$$x \xrightarrow{t2} y$$

for some accepting state y

MSC trace semantics for exit/entry states



Every MSC trace t can be split into pairs (t1,t2) where t1 leads to exit state.

For any t1 if there is a path

$$u \xrightarrow{t1} x$$

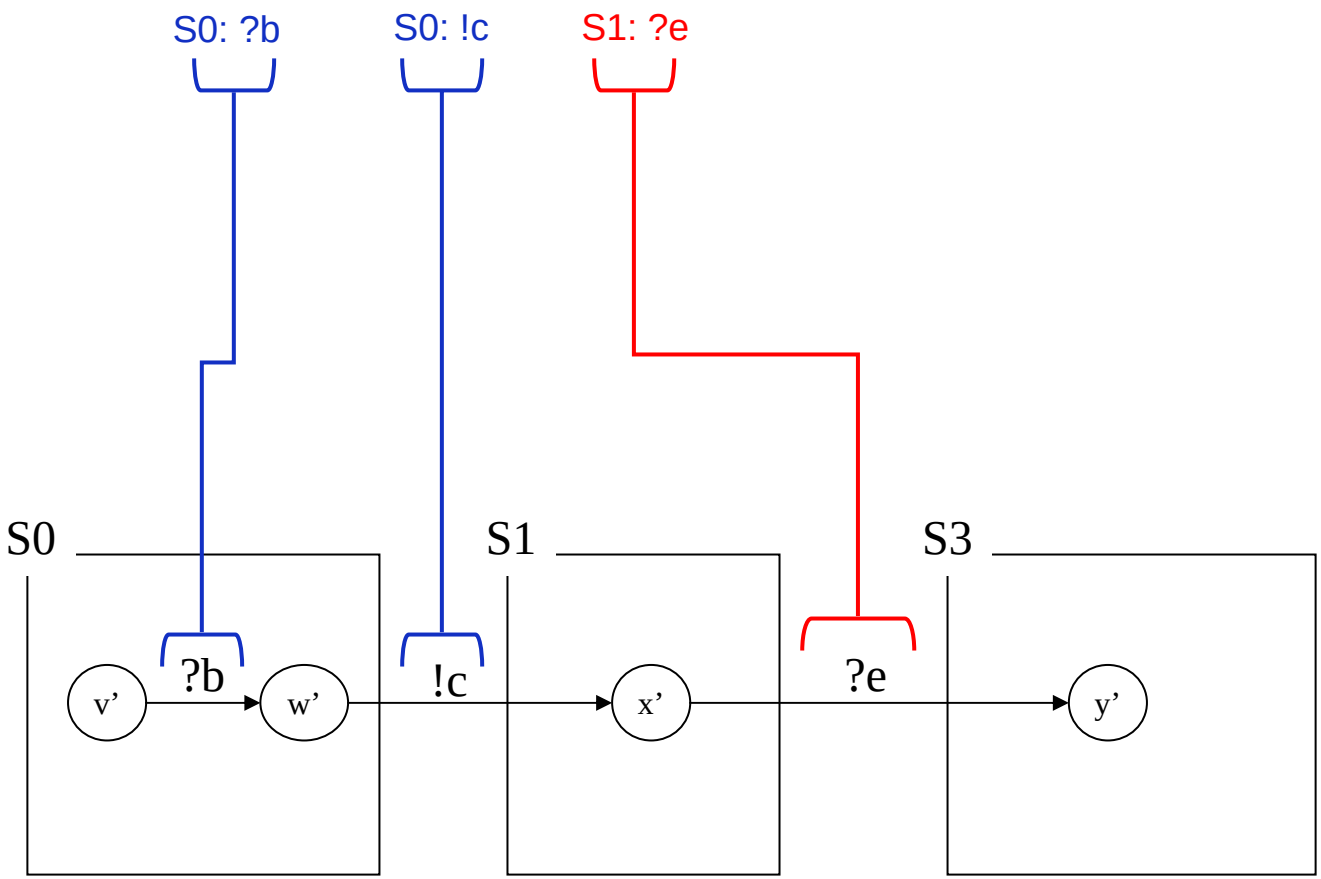
for any state u

then there is a path

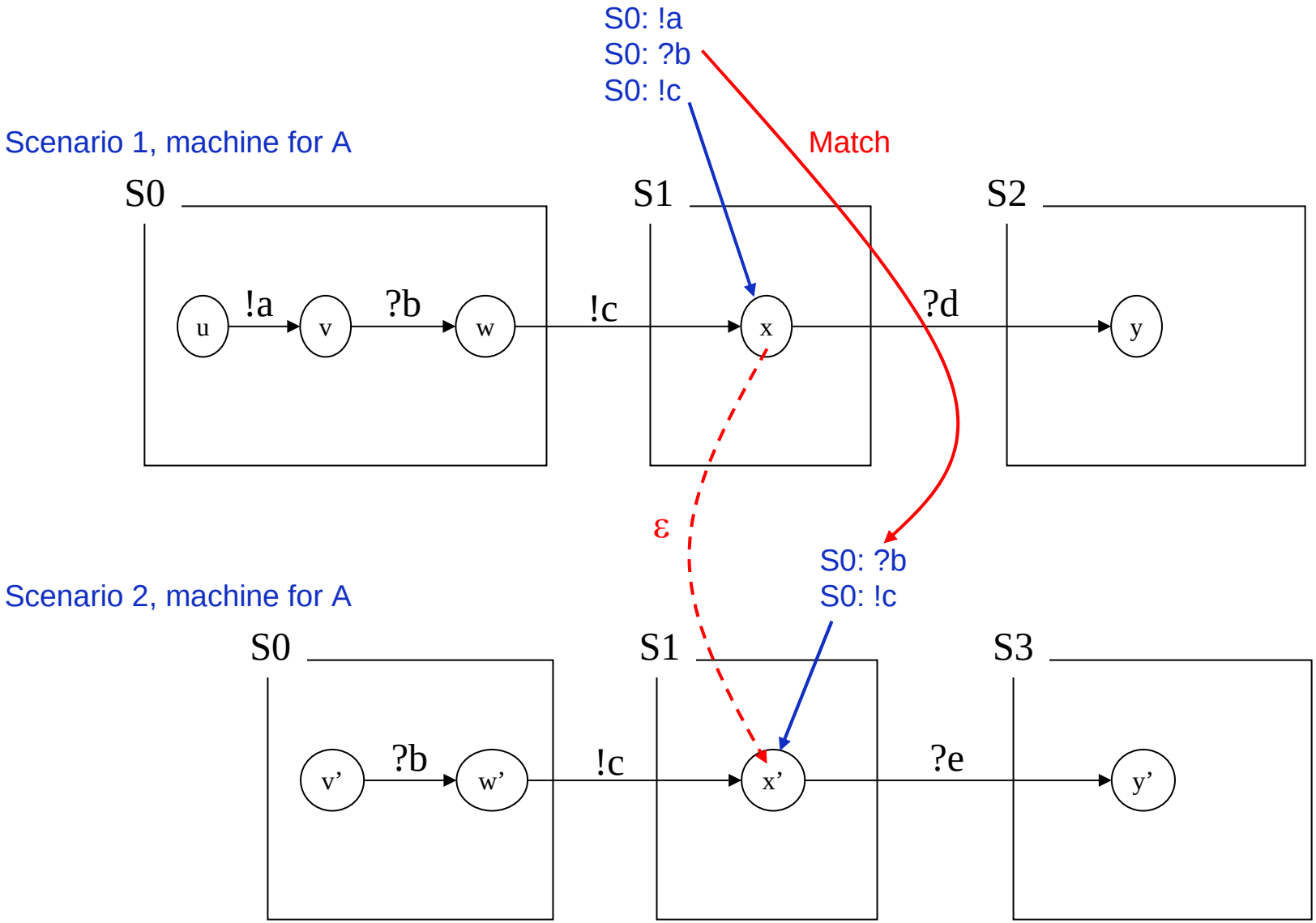
$$x \xrightarrow{t2} y$$

for some state y

State semantics



Overlapping Processes, continued



Overlapping Composition of Processes

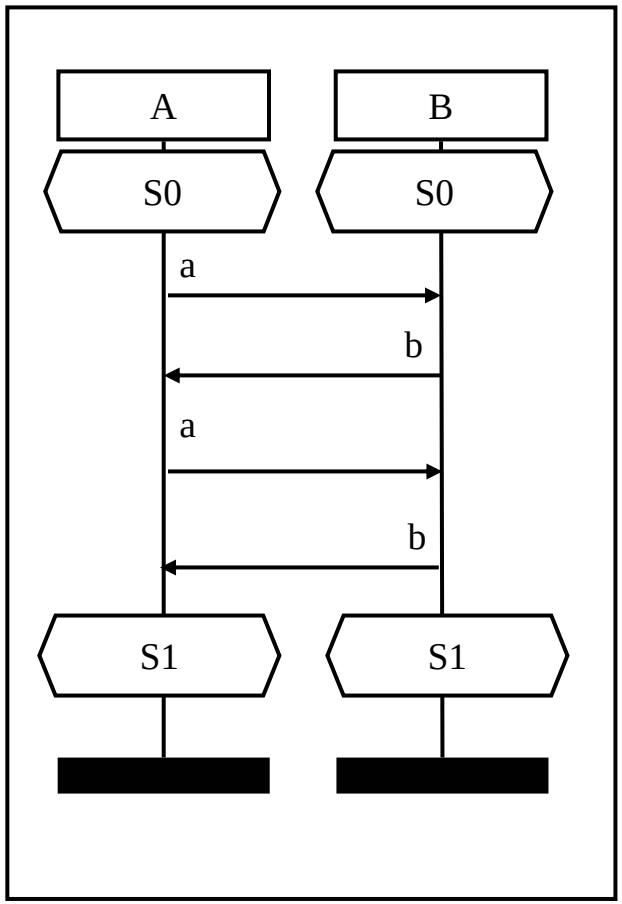
P trace simulates Q when:
given any (state annotated) execution traces t1 and t2:

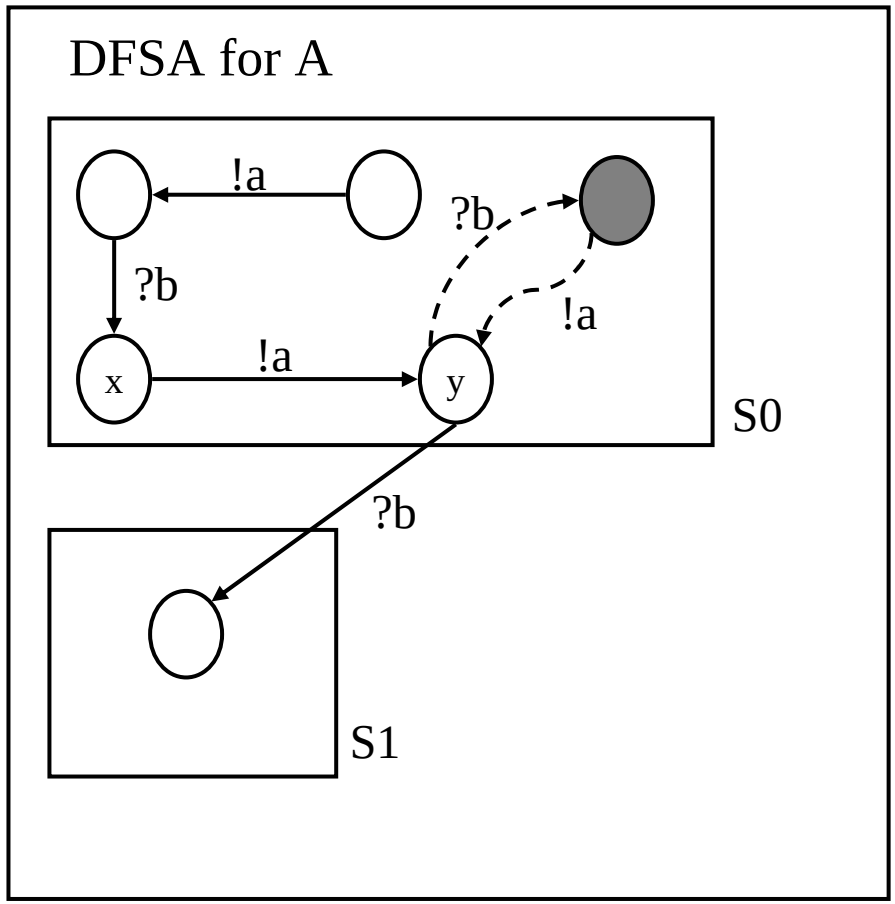
$$P \xrightarrow{t1} P1$$

$$Q \xrightarrow{t2} Q1$$

where t1 matches t2, then P1 must be able to simulate Q1

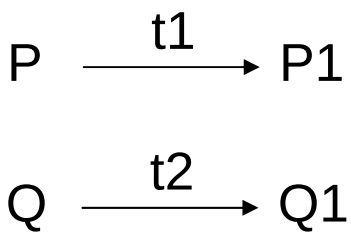
Livelock from naive composite state semantics





Exit State transition matching

P trace simulates Q when:
given any (state annotated) execution traces t1 and t2:



where t1 matches t2,
and t1, t2 have reached exit states
then P1 must be able to simulate Q1.

where t1 matches t2,
and t1, t2 have reached entry states
then P1 must be able to simulate Q1.

Temporal contexts for defining matching traces

LTL semantics for execution trace

$$\mathcal{X} \vdash (\|t\| \Rightarrow \Diamond \|t'\|)$$

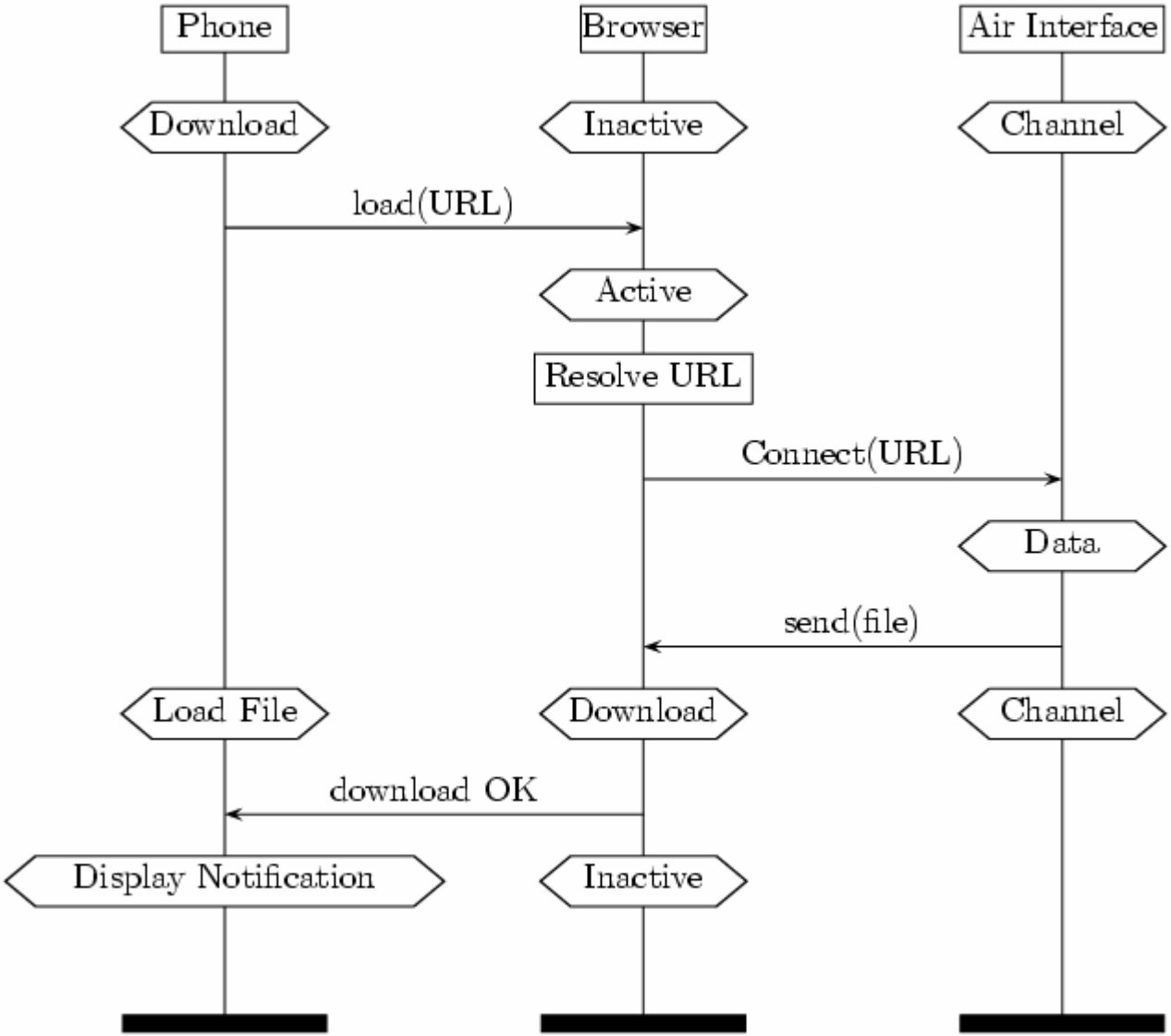
LTL formula defining context

$$\Box([\mathsf{load}(\mathsf{URL})](\mathsf{Active} \Rightarrow (\text{`Load File'} \ \mathcal{U} \ \mathsf{Inactive})))$$

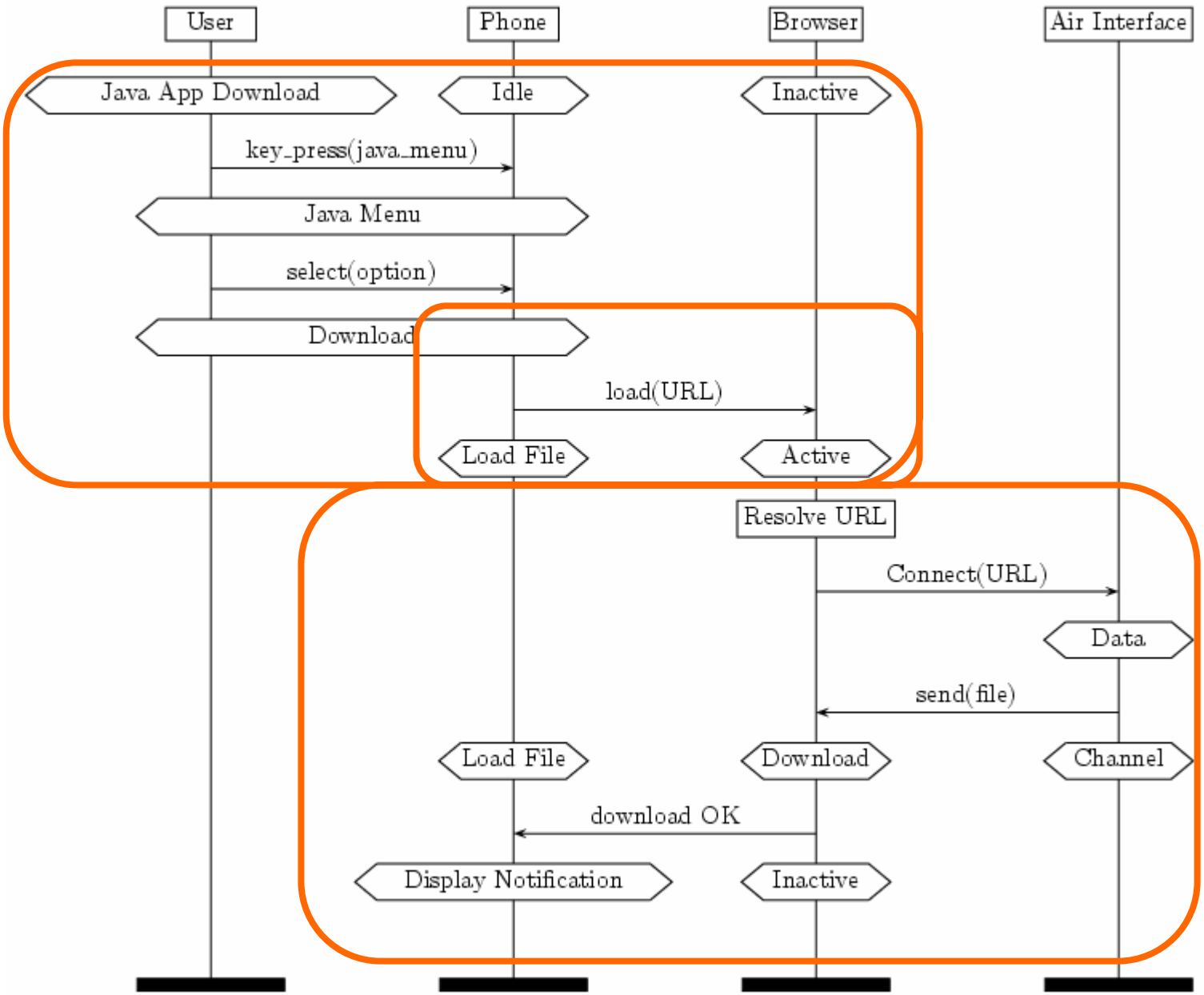
Event

Composite state

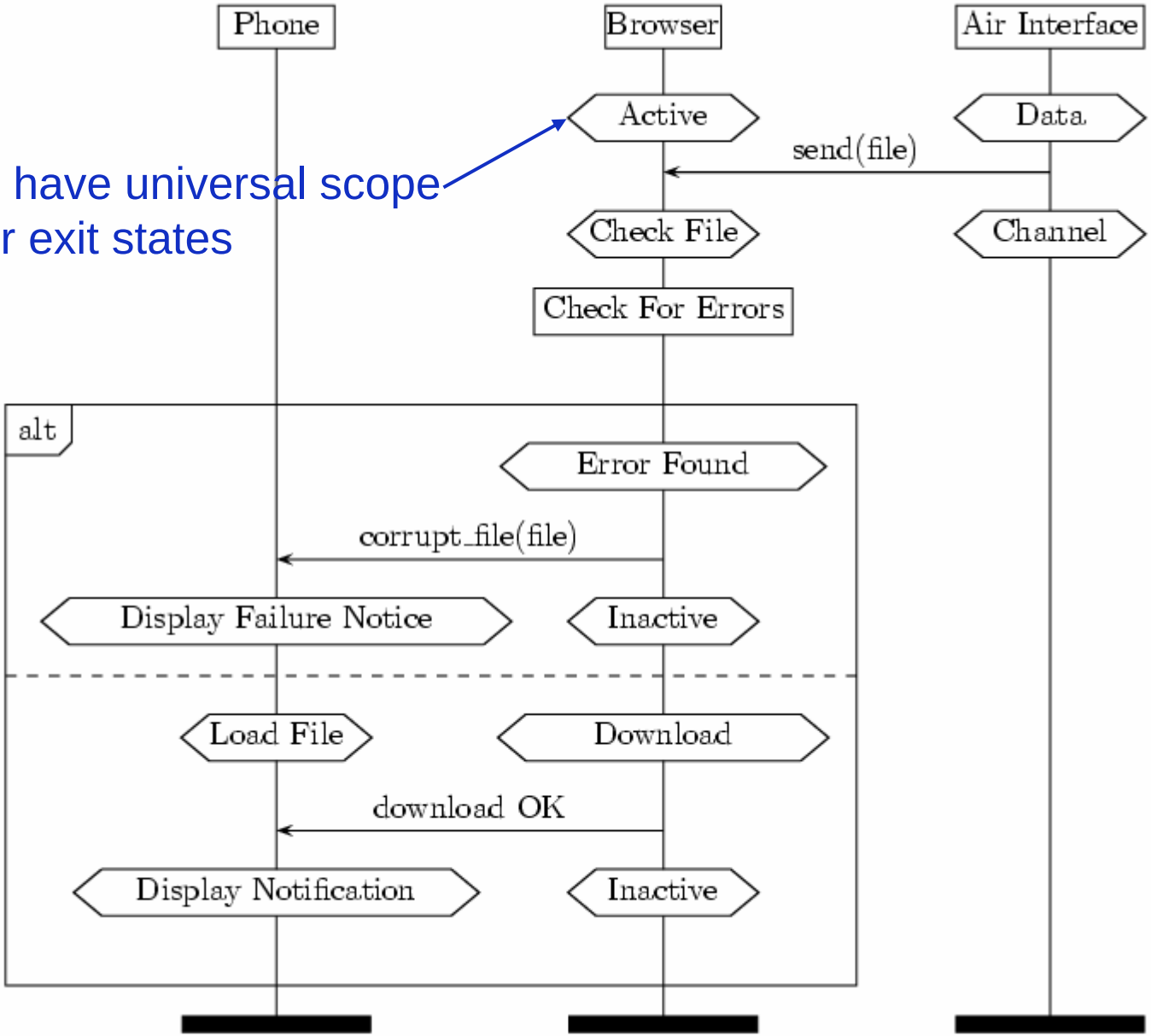
Download File with Browser



Overlap of Java Game and Browser Download

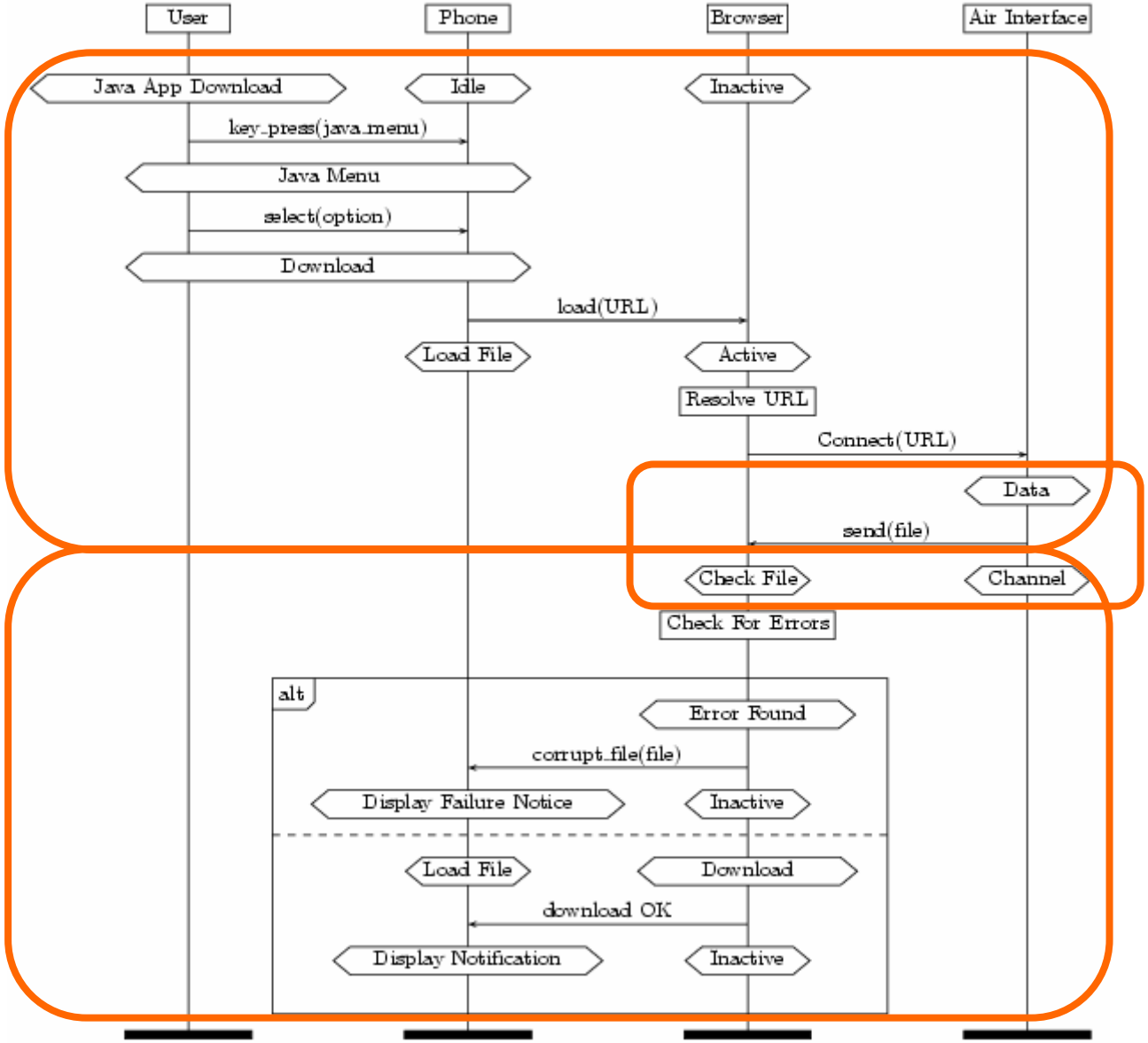


Error Check



Will have universal scope over exit states

Overlap Java App + Browser + Error Check



Questions