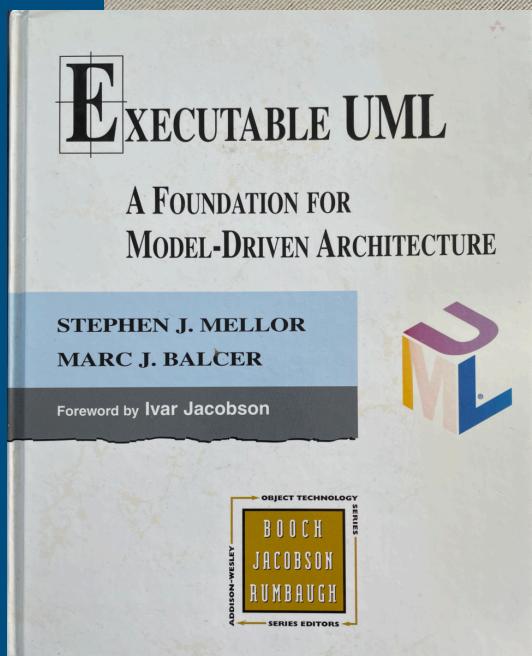
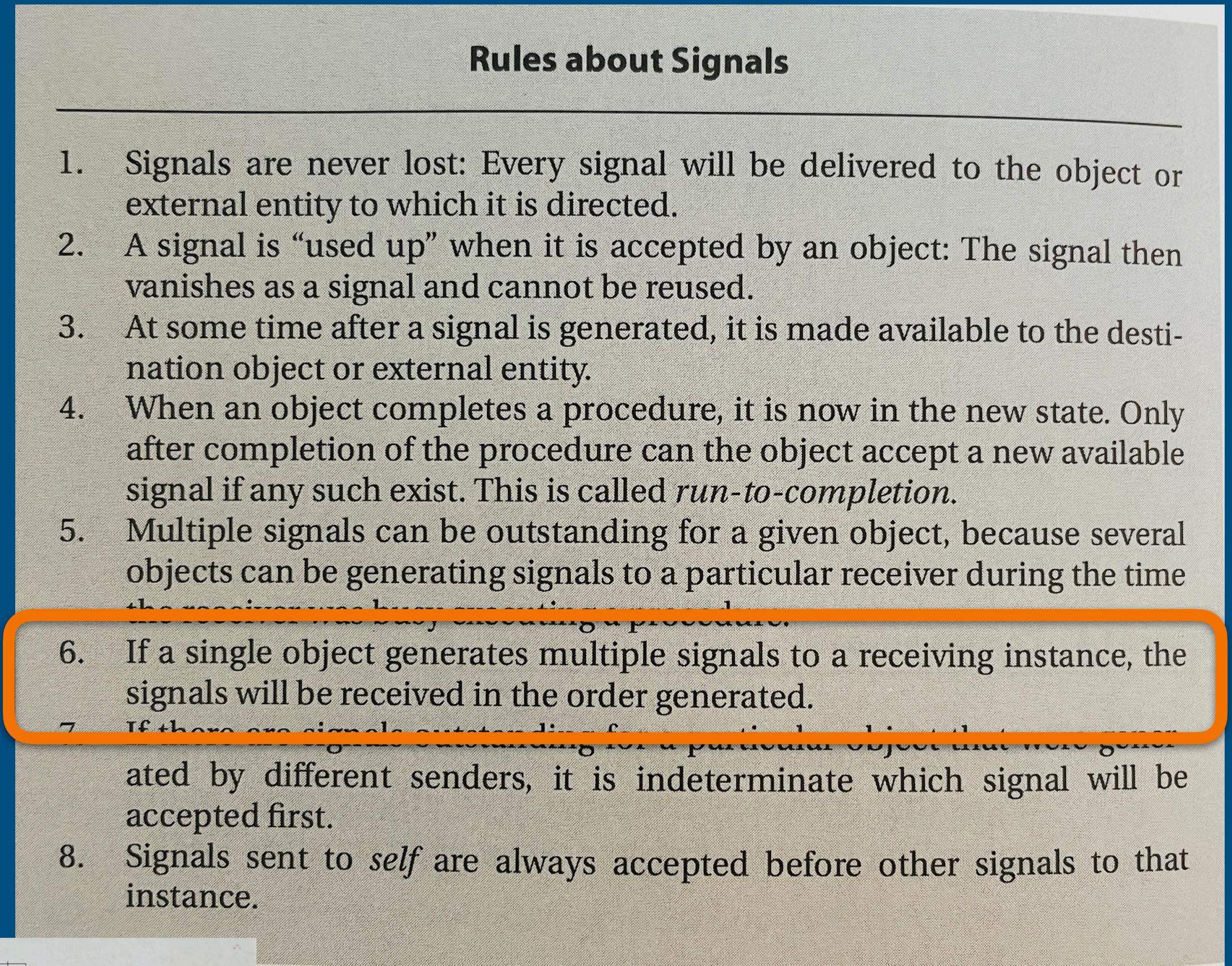


Self and non-self directed events

AN UNFORTUNATE CHOICE OF TERMS

CAREFULLY PARAPHRASING...



Executable UML: A Foundation for Model Driven Architecture,
Stephen Mellor, Mark Balcer, Addison-Wesley 2002

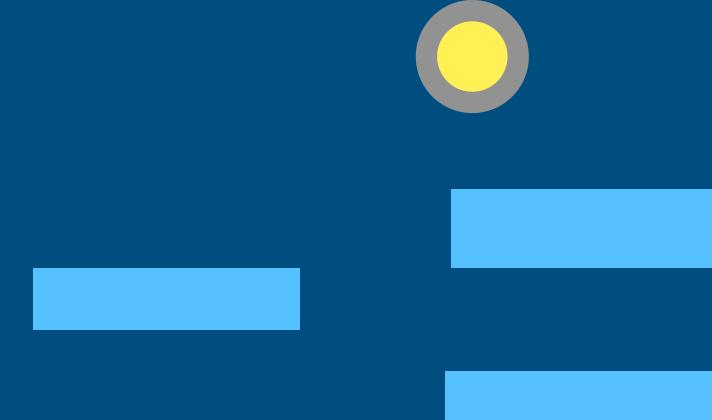
Events from the same **sender**
received in the order generated

No ordering otherwise...

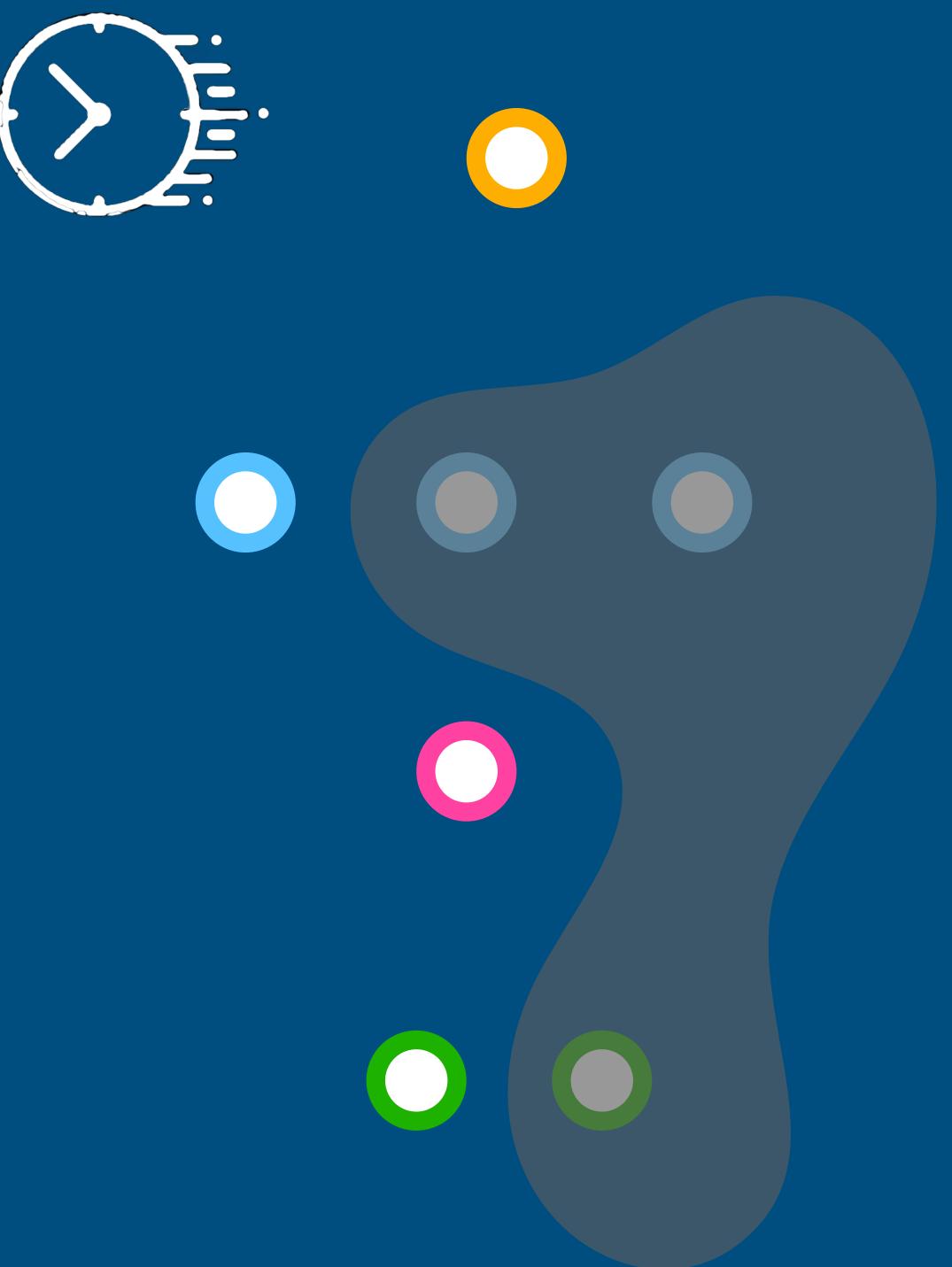
Except for self-directed events,
which are accepted before any
others

Pending events from different senders

Target instance
in some state



*Which event
can be accepted
next?*



Non-self
directed events
with partial
ordering

*Same color =
same sender*

*Cannot accept
these yet*

What about self directed events?

Are they ordered?

*Can they be delayed like
non-self directed events?*

*Can more than one be
pending at a time for the
same instance?*

*Might an instance send an
event to itself that should
NOT be treated like a self
directed event?*

Crazy examples of self directed events...

Instance of Flot tells all instances (including self) to move over at some point in the future

Move over -> Flot() @ shortly*

*Looking good -> me
Feeling great -> me
But kinda hungry -> me*

Multiple pending events to self?

Ordered?

Meaningful?

The original intention

This technique can be used to simplify state machines when a single (logical) state needs to be broken into several states, perhaps as the result of some conditional logic.

These signals are syntactically just like any other. However, signals to self are always treated first by the receiver, even if other events are outstanding.

UML has the notion of a *completion event* that serves the same purpose as a signal to self. Completion events are always handled first, just like signals to self. Completion events in UML are shown as an unlabeled transition.

The motivating, and possibly ONLY
meaningful use cases:

PROCEDURAL COMPLETION

initial computations complete -> me

CONDITIONAL COMPLETION

? too hot : cool down -> chill out -> me

If so...

Completion means nothing more than that an Activity has completed (under one or more mutually exclusive conditions)

And procedural completion is just a simple case of conditional completion (where there is only one condition)

So, for a given Instance, there can never be more than one completion event pending at one time

If so...

It's not about self vs. non-self directed instances

It's about logic completion vs. interaction

An Instance might complete its Activity

One Instance might interact with another Instance

And that Instance might be itself

So self vs non-self directed is not helpful terminology

What terminology WOULD be useful (precise)?

Shlaer-Mellor Metamodel / Execution Subsystem

