

defining **actions**

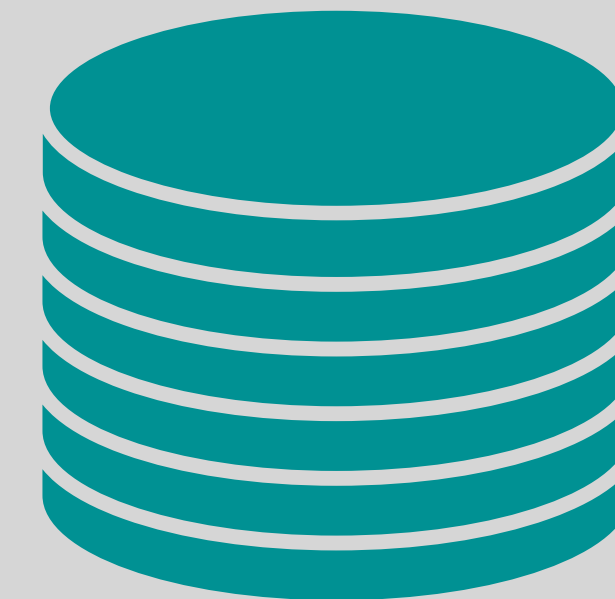
Daniel Jackson

how to describe
this for everyone
(so no UIs or code)



software systems
do things

what they do
affects what
they remember



software systems
remember things

what they
remember affects
what they do



behavior = a collection of scenarios

a scenario

create (Maido, 6:00pm, Jan-20-26): Slot_1

create (Maido, 7:30pm, Jan-20-26): Slot_2

reserve (Slot_1, Alice): Res_1

cancel (Res_1)

reserve (Slot_2, Alice): Res_2

redeem (Res_2)

relationships true at each point

at (Slot_2, Maido)

time (Slot_2, 7:30pm)

date (Slot_2, Jan-20-26)

for (Res_2, Slot_2)

by (Res_2, Alice)

individuals

Alice Bob

Maido Rosetta

Slot_1 Slot_2

Res_1

relationships

for (Res_1, Slot_2)

by (Res_1, Alice)

at (Slot_2, Maido)

actions

reserve (Slot_2, Alice): Res_1

cancel (Res_1)

create (Maido, 7:30pm, Jan-20-26): Slot_2

types of individuals

User

Restaurant

Slot

Reservation

types of relationships (relations)

for (Reservation, Slot)

by (Reservation, User)

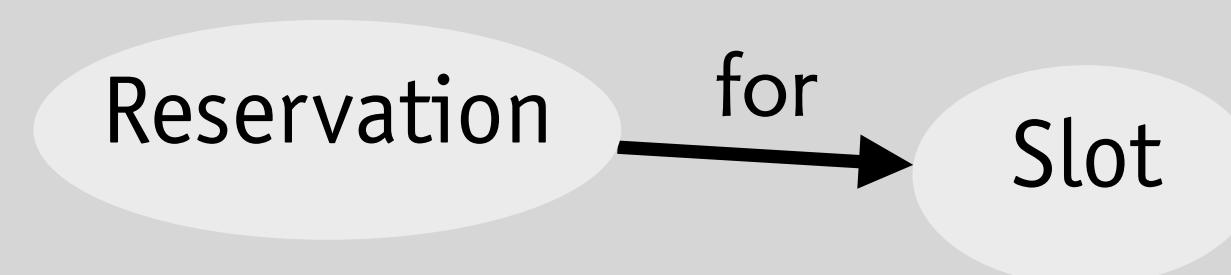
at (Slot, Restaurant)

types of actions

reserve (Slot, User): Reservation

cancel (Reservation)

create (Restaurant, Time, Date): Slot



how to define an action: the idea

give its name and
argument types

`reserve (Slot, User): Reservation`

introduce
variables

`reserve (s: Slot, u: User): (r: Reservation)`

say which
relationships
true or not before

not `for (_, s)`

say which
relationships
true or not after

`for (r, s) by (r, u)`

how to define an action: the notation

state

a set of Reservations with
a for Slot
a by User

actions

reserve (s: Slot, u: User): (r: Reservation)

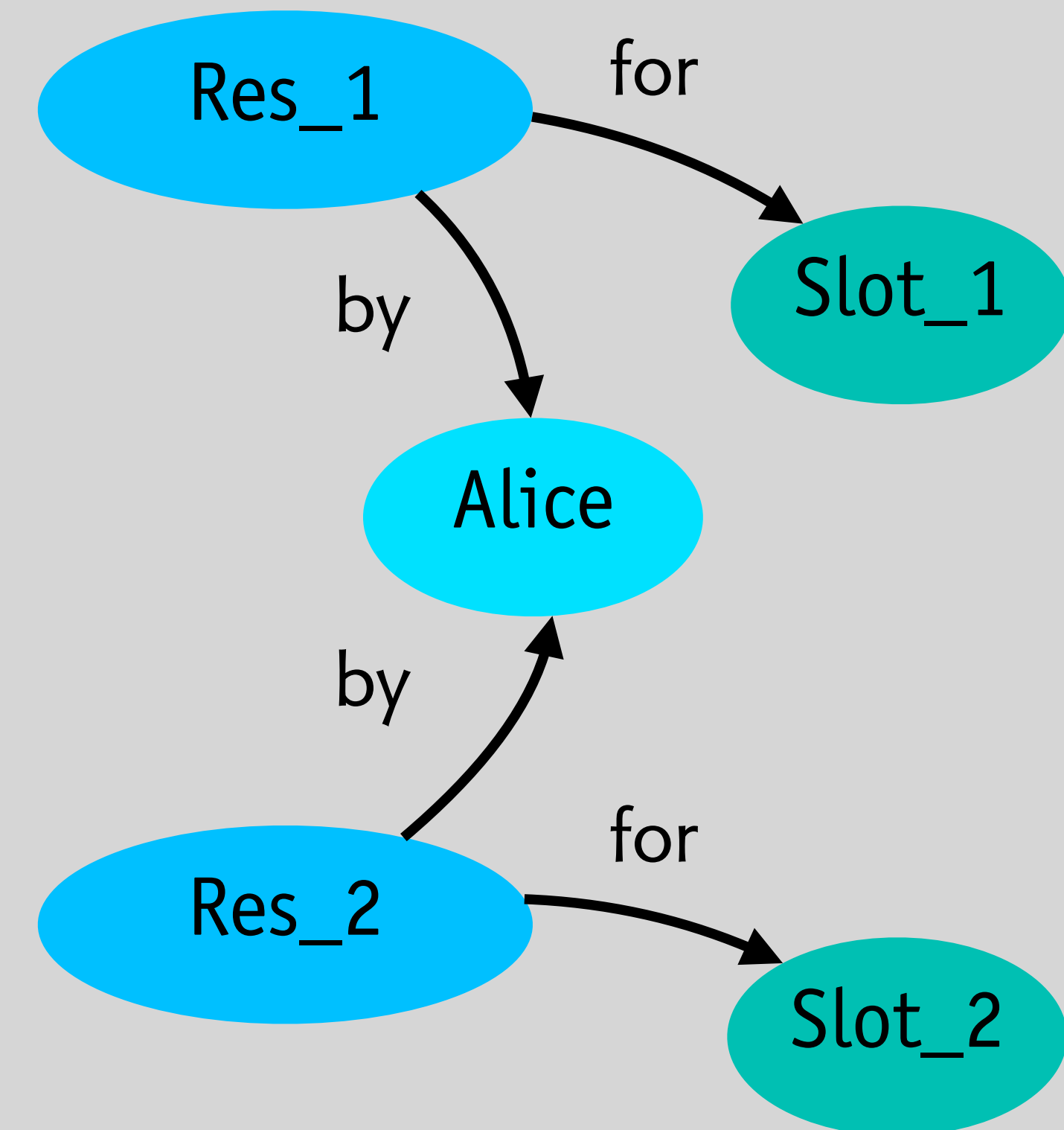
requires no existing reservation for s

ensures creates new reservation r that is for s and by u

cancel (r: Reservation)

requires r is an existing reservation

ensures removes r and its associations



`reserve (Slot_2, Alice): Res_2`

`cancel (Res_1)`

a surprise

individuals aren't objects!

concept Reserving

state

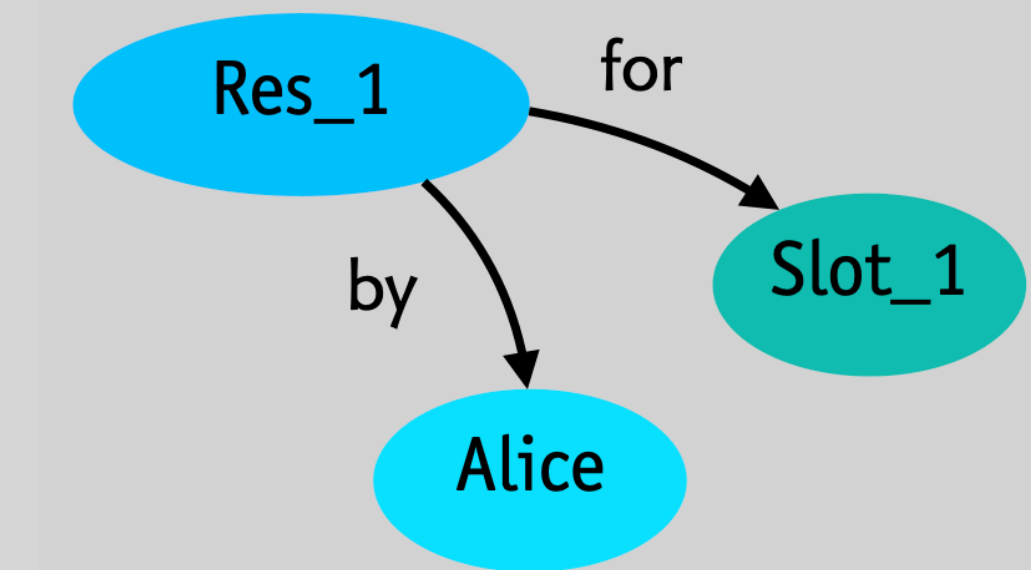
a set of Reservations with
a for Slot
a by User

notation just groups
relations together
to avoid repetition!

actions

reserve (s: Slot, u: User): (r: Reservation)

reserve returns a
reservation individual



does not return an
'object' that you can
look inside of

