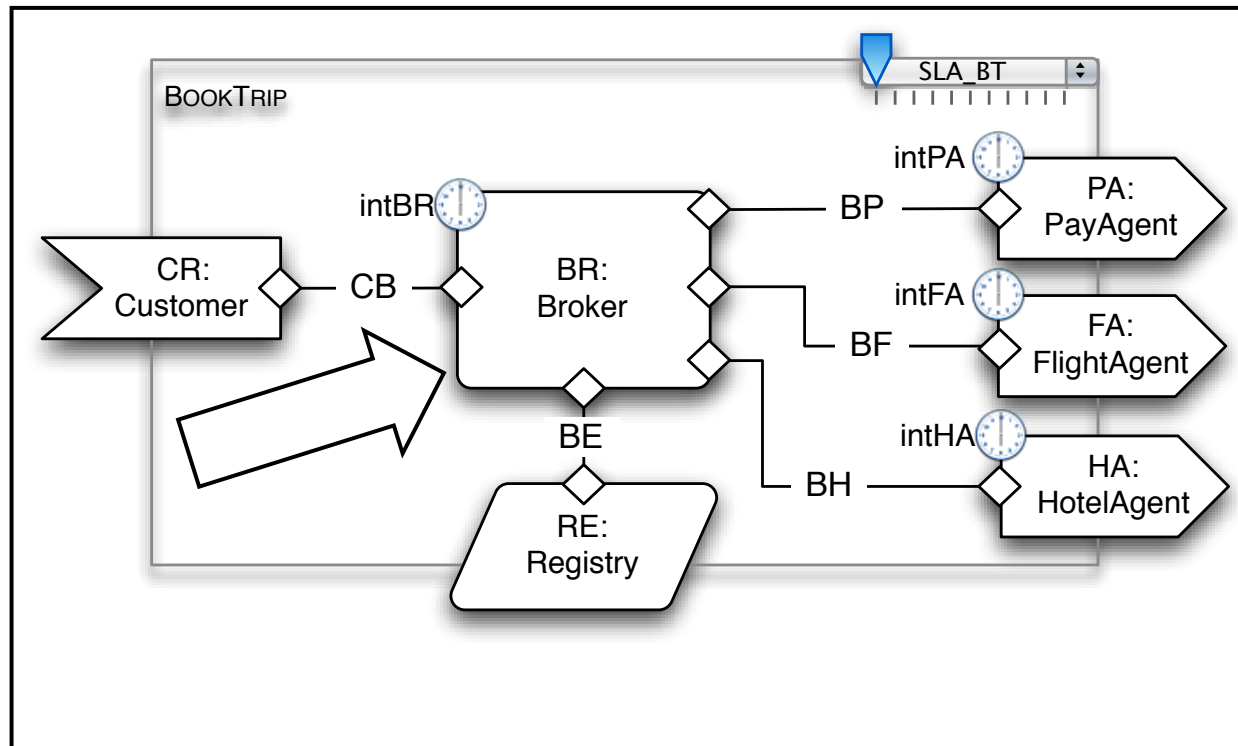


SRML Examples

The orchestration of the trip broker

The Module BookTrip



The States of Broker

BUSINESS ROLE Broker is

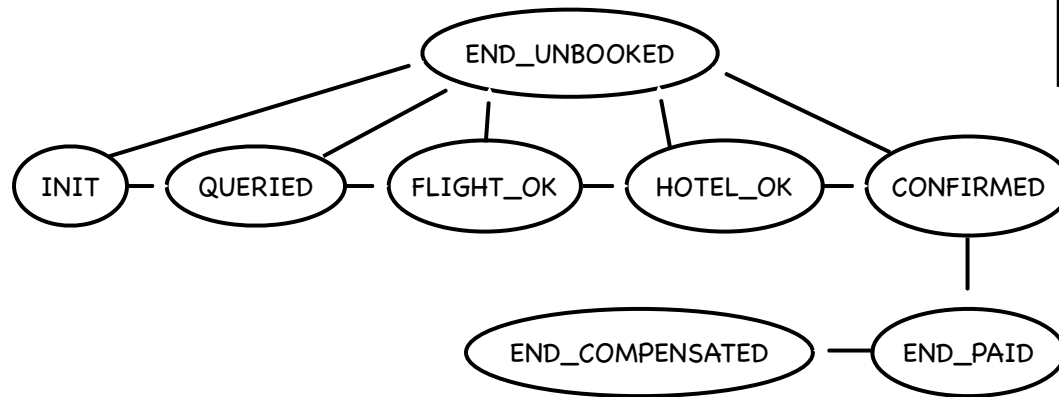
INTERACTIONS

...

ORCHESRATION

local

s:[INIT, QUERIED, FLIGHT_OK, HOTEL_OK,
CONFIRMED, END_PAID, END_UNBOOKED, END_COMPENSATED]

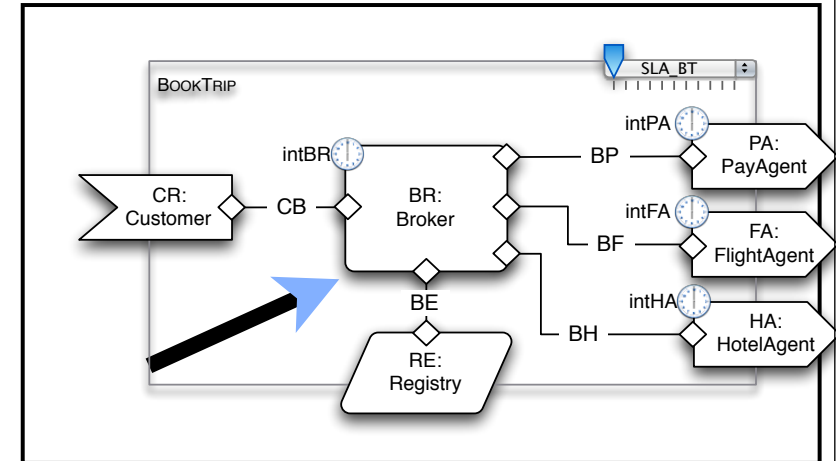


intBR init

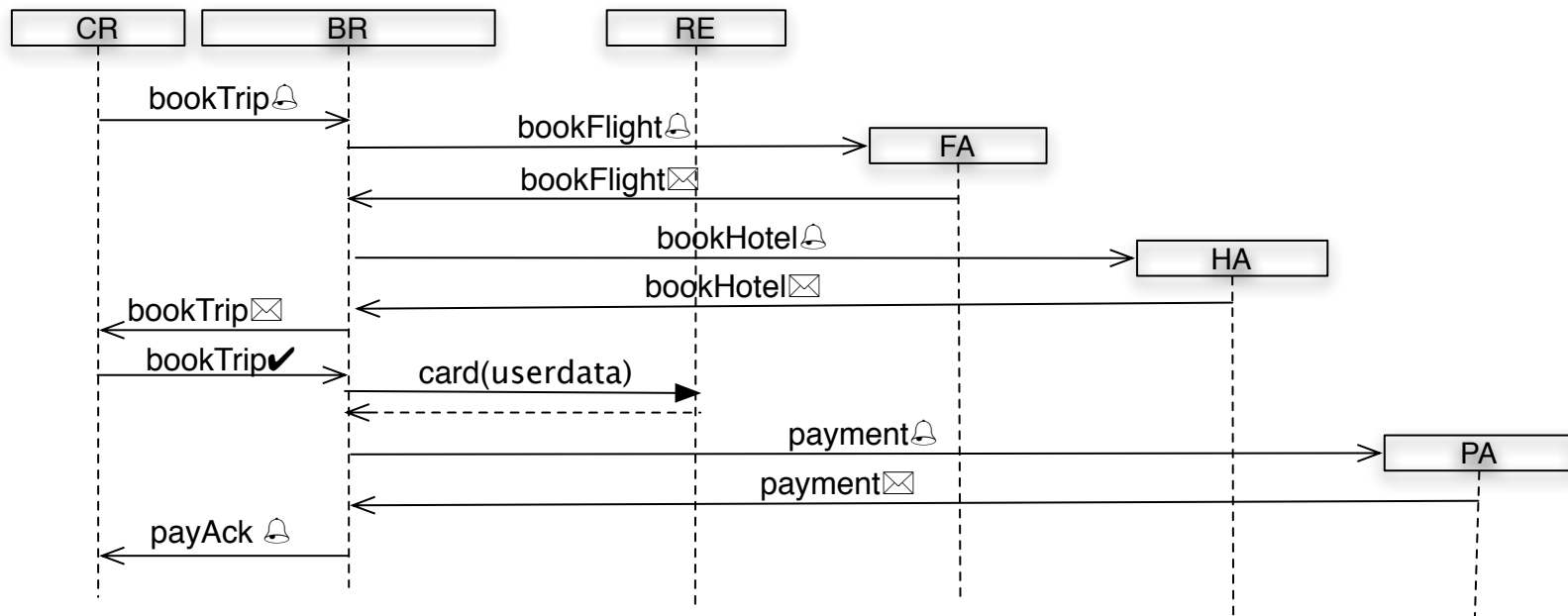
s=INIT

intBR term

s=END_UNBOOKED ∨
(s=END_PAID ∧ today ≥ bookTrip.out) ∨
END_COMPENSATED



One Sequence Diagram...



Interactions

BUSINESS ROLE Broker **is**

INTERACTIONS

r&s bookTrip

🔔 from,to:airport
out,in:date
usr:userdata

✉ fconf:fcode
hconf:hcode
amount:moneyData

s&r bookFlight

🔔 from,to:airport
out,in:date

✉ fconf:fcode
amount:moneyData
beneficiary:accountNo
payService:serviceId

...

Transitions

...

transition Request

triggeredBy bookTrip🔔

guardedBy s=INIT

effects bookTrip.out≤today ⊃ s'=END_UNBOOKED

∧ bookTrip.out>today ⊃ s'=QUERIED

sends bookTrip.out≤today ⊃ bookTrip✉

∧ bookTrip.Reply=false

∧ bookTrip.out>today ⊃ bookFlight🔔

∧ bookFlight.from=bookTrip.from

∧ bookFlight.to=bookTrip.to

∧ bookFlight.out=bookTrip.out

∧ bookFlight.in=bookTrip.in

...

Interactions

BUSINESS ROLE Broker **is**

INTERACTIONS

r&s bookTrip

🔔 from,to:airport
out,in:date
usr:userdata

✉ fconf:fcode
hconf:hcode
amount:moneyData

s&r bookFlight

🔔 from,to:airport
out,in:date

✉ fconf:fcode
amount:moneyData
beneficiary:accountNo
payService:serviceId

s&r bookHotel

🔔 checkin,checkout:date
traveller:usrdata
✉ amount:moneyData
hconf:hcode

...

Transitions

...

transition FlightAnswer

triggeredBy bookFlight✉

guardedBy s=QUERIED

effects bookFlight.Reply=true \supset s'=FLIGHT_OK

\wedge bookFlight.Reply=false \supset s'=END_UNBOOKED

sends bookFlight.Reply=true \supset bookHotel🔔

\wedge bookHotel.checkin=bookTrip.out

\wedge bookHotel.checkout=bookTrip.in

\wedge bookHotel.traveller=bookTrip.usr

\wedge bookFlight.Reply=false \supset bookTrip✉

\wedge bookTrip.Reply=false

...

Interactions

BUSINESS ROLE Broker is

INTERACTIONS

r&s bookTrip

🔔 from,to:airport
out,in:date
usr:userdata

✉ fconf:fcode
hconf:hcode
amount:moneyData

s&r bookFlight

🔔 from,to:airport
out,in:date

✉ fconf:fcode
amount:moneyData
beneficiary:accountNo
payService:serviceId

s&r bookHotel

🔔 checkin,checkout:date
traveller:usrdata

✉ amount:moneyData
hconf:hcode

...

Transitions

...

transition HotelAnswer

triggeredBy bookHotel✉

guardedBy s=FLIGHT_OK

effects bookHotel.Reply \supset s'=HOTEL_OK

\wedge \neg bookHotel.Reply \supset s'=END_UNBOOKED

sends bookHotel.Reply \supset bookTrip✉

\wedge bookTrip.Reply=true

\wedge bookTrip.fconf=bookFlight.fconf

\wedge bookTrip.amount=bookFlight.amount+
bookHotel.amount

\wedge bookTrip.hconf=bookHotel.hconf

\wedge \neg bookHotel.Reply \supset bookFlight✕

\wedge bookTrip✉

\wedge bookTrip.Reply=false

...

Find the difference ... (there are 5!)

```
...
transition HotelAnswer1
  triggeredBy bookHotel☒
  guardedBy s=FLIGHT_OK
  effects bookHotel.Reply ⊃ s'=HOTEL_OK
    ∧ ¬bookHotel.Reply ⊃ s'=END_UNBOOKED
  sends bookHotel.Reply ⊃ bookTrip☒
    ∧ bookTrip.Reply=true
    ∧ bookTrip.fconf=bookFlight.fconf
    ∧ bookTrip.amount=bookFlight.amount+
      bookHotel.amount
    ∧ bookTrip.hconf=bookHotel.hconf
  ∧ ¬bookHotel.Reply ⊃ bookFlight✗
    ∧ bookTrip☒
    ∧ bookTrip.Reply=false
...
```

```
...
transition HotelAnswer2
  triggeredBy bookHotel☒
  guardedBy s=FLIGHT_OK
  effects bookHotel.Reply=true ⊃ s'=HOTEL_OK
    ∧ bookHotel.Reply=false ⊃ s'=END_UNBOOKED
  sends s'=HOTEL_OK ⊃ bookTrip☒
    ∧ bookTrip.Reply=true
    ∧ bookTrip.fconf=bookFlight.fconf
    ∧ bookTrip.amount=bookFlight.amount+
      bookHotel.amount
    ∧ bookTrip.hconf=bookHotel.hconf
  ∧ s'=END_UNBOOKED ⊃ bookFlight✗
    ∧ bookTrip☒
    ∧ bookTrip.Reply=false
...
```

Is the transition on the right correct? YES!

Are the two transitions equivalent? YES!

Find the difference ...

```
...
transition HotelAnswer1
  triggeredBy bookHotel☒
  guardedBy s=FLIGHT_OK
  effects bookHotel.Reply ⊃ s'=HOTEL_OK
    ∧ ¬bookHotel.Reply ⊃ s'=END_UNBOOKED
  sends bookHotel.Reply ⊃ bookTrip☒
    ∧ bookTrip.Reply=true
    ∧ bookTrip.fconf=bookFlight.fconf
    ∧ bookTrip.amount=bookFlight.amount+
      bookHotel.amount
    ∧ bookTrip.hconf=bookHotel.hconf
  ∧ ¬bookHotel.Reply ⊃ bookFlight✗
    ∧ bookTrip☒
    ∧ bookTrip.Reply=false
...
```

```
...
transition HotelAnswer1
  triggeredBy bookHotel☒
  guardedBy s=FLIGHT_OK
  effects ¬bookHotel.Reply ⊃ s'=END_UNBOOKED
    ∧ bookHotel.Reply ⊃ s'=HOTEL_OK
  sends ¬bookHotel.Reply ⊃ bookFlight✗
    ∧ bookTrip☒
    ∧ bookTrip.Reply=false
  ∧ bookHotel.Reply ⊃ bookTrip☒
    ∧ bookTrip.fconf=bookFlight.fconf
    ∧ bookTrip.Reply=true
    ∧ bookTrip.amount=bookFlight.amount+
      bookHotel.amount
    ∧ bookTrip.hconf=bookHotel.hconf
...
```

Are the two transitions equivalent?

YES!

Find the 5 differences ... (last game!)

```
...
transition HotelAnswer1
  triggeredBy bookHotel☒
  guardedBy s=FLIGHT_OK
  effects bookHotel.Reply  $\supset$  s'=HOTEL_OK
     $\wedge$   $\neg$ bookHotel.Reply  $\supset$  s'=END_UNBOOKED
  sends bookHotel.Reply  $\supset$  bookTrip☒
     $\wedge$  bookTrip.Reply=true
     $\wedge$  bookTrip.fconf=bookFlight.fconf
     $\wedge$  bookTrip.amount=bookFlight.amount+
      bookHotel.amount
     $\wedge$  bookTrip.hconf=bookHotel.hconf
   $\wedge$   $\neg$ bookHotel.Reply  $\supset$  bookFlight✗
     $\wedge$  bookTrip☒
     $\wedge$  bookTrip.Reply=false
  ...
```

```
...
transition HotelAnswer1
  triggeredBy bookHotel☒
  guardedBy FLIGHT_OK
  effects bookHotel.Reply  $\supset$  HOTEL_OK
     $\wedge$   $\neg$ bookHotel.Reply  $\supset$  s'=END_UNBOOKED
  sends bookHotel.Reply  $\supset$  bookTrip☒
     $\wedge$  bookTrip.Reply
     $\wedge$  bookTrip.fconf=bookFlight.fconf
     $\wedge$  bookTrip.amount=bookFlight.amount+
      bookHotel.amount
     $\wedge$  bookTrip.hconf=bookHotel.hconf
   $\wedge$   $\neg$ bookHotel.Reply  $\supset$  bookFlight✗
  [  $\wedge$  bookTrip☒
     $\wedge$   $\neg$ bookTrip.Reply'
  ]
  ...
```

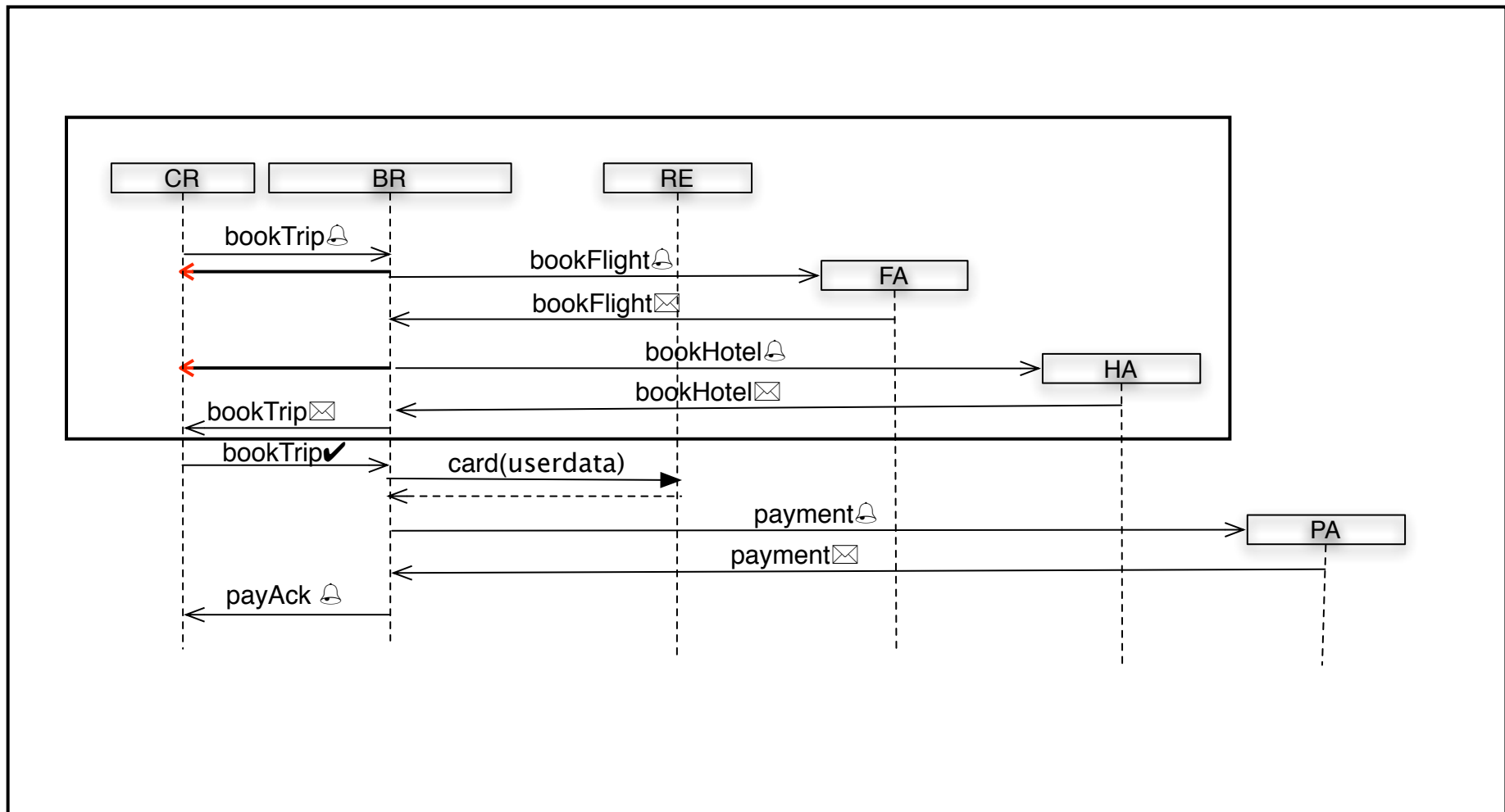
Is the transitions on the right correct?

NO!

Are the two transitions equivalent?

NO!

One Sequence Diagram...



Interactions

BUSINESS ROLE Broker is

INTERACTIONS

...

s&r bookFlight

🔔 from,to:airport
out,in:date

✉ fconf:fcode
amount:moneyData
payService:serviceId

s&r bookHotel

🔔 checkin,checkout:date
traveller:usrdata

✉ amount:moneyData
hconf:hcode

s&r payment

🔔 amount:moneyvalue
originator:usrdata
cardNo:paydata

✉ proof:pcode

ask card(userdata):paydata

Write a transition that..

- Has name Commitment
- It is triggered by the receive of the commit-event of bookTrip occurring in the state s=HOTEL_OK
- Goes in the state CONFIRMED
- Triggers the initiation-event of payment (the amount is the sum of the flight and of the hotel price, the card number is obtained through the synchronous interaction card)

Commitment

transition Commitment

triggeredBy bookTrip✓

guardedBy s=HOTEL_OK

effects s'=CONFIRMED

sends payment🔔

^ payment.amount=bookFlight.amount+bookHotel.amount

^ payment.originator=bookTrip.usr

^ payAck.cardNo=card(bookTrip.usr)

Interactions

BUSINESS ROLE Broker is

INTERACTIONS

r&s bookTrip

🔔 from,to:airport

out,in:date

usr:userdata

✉ fconf:fcode

hconf:hcode

amount:moneyData

s&r bookFlight

🔔 from,to:airport

out,in:date

✉ fconf:fcode

amount:moneyData

payService:serviceId

...

Write a transition that..

- ⦿ Has name Timeout
- ⦿ It is triggered by the event
now>bookTrip.Useby occurring in the
state s=HOTEL_OK
- ⦿ Goes in the state END_UNBOOKED

Timeout

```
transition Timeout  
  triggeredBy now>bookTrip.UseBy  
  guardedBy s=HOTEL_OK  
  effects s'=END_UNBOOKED
```

Interactions

BUSINESS ROLE Broker is

INTERACTIONS

...

s&r payment

🔔 amount:moneyvalue
originator:usrdata
cardNo:paydata

✉ proof:pcode

snd payAck

🔔 proof:pcode
status:boolean

s&r bookHotel

🔔 checkin,checkout:date
traveller:usrdata

✉ amount:moneyData
hconf:hcode

ask card(userdata):paydata

Write a transition that..

- Has name PaymentAnswer
- It is triggered by the receive-event payment occurring in the state S=CONFIRMED
 - if the parameter Reply is true than the transition goes in the state s=END_PAID
 - otherwise the transition goes in state S=END_UNBOOKED
- In both cases the transition triggers the initiation events for payAck setting the parameter status of payAck as the parameter Reply of payment and
 - if Reply of payment of payment is true then set pcode os payAck as the parameter pcode of payment

PaymentAnswer

```
transition PaymentAnswer
  triggeredBy payment☒
  guardedBy s=CONFIRMED
  effects payment.Reply  $\supset$  s'=END_PAID
   $\wedge$   $\neg$ payment.Reply  $\supset$  s'=END_UNBOOKED
  sends payAck🔔
   $\wedge$  payAck.status=payment.Reply
   $\wedge$  payment.Reply  $\supset$  payAck.proof=payment.proof
```

Interactions

BUSINESS ROLE Broker **is**

INTERACTIONS

r&s bookTrip

🔔 from,to:airport
out,in:date
usr:userdata

✉ fconf:fcode
hconf:hcode
amount:moneyData

s&r bookFlight

🔔 from,to:airport
out,in:date

✉ fconf:fcode
amount:moneyData
beneficiary:accountNo
payService:serviceId

s&r bookHotel

🔔 checkin,checkout:date
traveller:usrdata

✉ amount:moneyData
hconf:hcode

...

Write a transition that..

- 🔔 Has name TripCompensate
- 🔔 It is triggered by the revoke-event bookTrip occurring in the state END_PAID ONLY IF today is prior than the parameter out of bookTrip
- 🔔 Goes in the state END_COMPENSATED
- 🔔 Triggers the revoke (compensation) events for bookFlight and bookHotel

TripCompensate

TripCompensate

triggeredBy bookTrip†

guardedBy s=END_PAID \wedge today<bookTrip.out

effects s'=END_COMPENSATED

sends bookFlight† \wedge bookHotel†