SRML

exercise3: the specifications
The Orchestration of Travel Booking

**CR:** Customer

**BA:** BookingAgent

**HA:** HotelAgent

**PA:** PayAgent

**FA:** FlightAgent

**BD:** UsrDB
Complete the following business protocol...

**BUSINESS PROTOCOL** PayAgent is

**INTERACTIONS**

**INTERACTIONS**

**r&s payment**
- amount:moneyvalue
- beneficiary:usrdata
- originator:usrdata
- cardNo:paydata
- proof:pcode

**snd payNotify**
- status:bool

Define the following properties for PayAgent:

- in the initial state the pay agent is ready to engage in the initiation event of payment
- the pay agent will eventually notify the customer (with the initiation event of payNotify) after replying to the booking agent with the reply event of payment
Solution

BUSINESS PROTOCOL PayAgent is

INTERACTIONS
r&s payment
  amount:moneyvalue
  beneficiary:usrdata
  originator:usrdata
  cardNo:paydata
  proof:pcode
snd payNotify
  status:bool

BEHAVIOUR
initiallyEnabled payment?
payment! ensures payNotify!

ONLY FOR CONVERSATIONAL INTERACTIONS

payment? ensures payment!
payment! ∧ payment.Reply enables paymentx?
until payment✓? or payment.UseBy>=now

payment! ∧ payment.Reply enables payment✓?
until paymentx? or payment.UseBy>=now

payment✓? enables payment??
Define the following properties for HotelAgent

- in the initial state the hotel agent is ready to engage in the initiation event of lockHotel
- once the confirmation-event of lockHotel is received, a revoke event for lockHotel is possible before the date of arrival (represented by the parameter checkin of lockHotel)
Solution

BUSINESS PROTOCOL HotelAgent is

INTERACTIONS

r&s lockHotel

checkin,checkout:date

name:usrdata

hconf:hcode

BEHAVIOUR

initiallyEnabled lockHotel?

lockHotel✓? enables lockHotel‡? until today≥lockHotel.checkin
Complete the following business protocol...

**BUSINESS PROTOCOL** FlightAgent is

**INTERACTIONS**

_**r&s** lockFlight_  
 Depository  
 from,to:airport,  
 out,in:date,  
 traveller:usrData  

_**r&s** lockFlight_  
 Repository  
 fconf:fcode,  
 hconf:hcode,  
 amount:moneyValue,  
 beneficiary:accountNo,  
 payService:serviceId

_**rcv** payAck_  
 Depository  
 proof:pcode  
 status:boolean

_**snd** payRefund_  
 Depository  
 amount:moneyVal
Solution

Business Protocol: FlightAgent is

Interactions:

r&s lockFlight
  from, to: airport,
  out, in: date,
  traveller: userData
  fconf: fcode,
  hconf: hcode,
  amount: moneyValue,
  beneficiary: accountNo,
  payService: serviceId

rcv payAck
  proof: pcode
  status: boolean

snd payRefund
  amount: moneyValue

- in the initial state the flight agent is ready to engage in the initiation event of lockFlight
- the receive event of PayAck will be (for sure!) executed after and only after the occurrence of the reply event of lockFlight with positive reply
- the revoke of lockFlight will be accepted after and only after the send of payAck with positive status if today is earlier than 7 days (7th day included) before the date of parameter lockFlight.out
- the initiation event of payRefund will occur upon but not before receiving the commit of lockFlight
BUSINESS PROTOCOL FlightAgent is

INTERACTIONS

\[ \text{r&s lockFlight} \]

\[ \text{rcv payAck} \]

\[ \text{snd payRefund} \]

\[ a \text{ from,to:airport,} \]
\[ a \text{ out,in:date,} \]
\[ a \text{ traveller:usrData} \]
\[ a \text{ fconf:fcode,} \]
\[ a \text{ hconf:hcode,} \]
\[ a \text{ amount:moneyValue,} \]
\[ a \text{ beneficiary:accountNo,} \]
\[ a \text{ payService:serviceId} \]

\[ a \text{ proof:pcode} \]
\[ a \text{ status:boolean} \]

\[ a \text{ amount:moneyVal} \]

BEHAVIOUR

\[ \text{initiallyEnabled lockFlight}\? \]

\[ \text{lockFlight}\! \land \text{lockFlight.Reply enables payAck}\? \]

\[ \text{payAck}\? \land \text{payAck.status enables lockFlight}\! \]

\[ \text{lockFlight}\! \land \text{lockFlight.out>=today+7 ensures payRefund}\! \]

\[ \text{lockFlight}\! \land \text{lockFlight.out<today+7 ensures payRefund}\! \]
BUSINESS PROTOCOL FlightAgent is

INTERACTIONS

r&s lockFlight

| from, to: airport, |
| out, in: date, |
| traveller: usrData |

| fconf: fcode, |
| hconf: hcode, |
| amount: moneyValue, |
| beneficiary: accountNo, |
| payService: serviceId |

rcv payAck

| proof: pcode |
| status: boolean |

snd payRefund

| amount: moneyVal |

Is the following property guaranteed by the computational model?

payAck?$ ensures payAck?! NO!