

Problems

EXTERNAL POLICY

<[0..1],max,min,0,1>

SLA VARIABLES

HA.DIST2CENTRE, HA.DIST2METRO, FA.BOOKFEE, CR.BOOKFEE, CR.PERC, FA.PERC

CONSTRAINTS

...

C₂: {CR.BOOKFEE} $def_2(n) = \text{if } n > 5 \text{ then } 1 \text{ otherwise } 0$

C₃: {CR.BOOKFEE, FA.BOOKFEE} $def_3(d,p) = \text{if } d > p \text{ then } 1 - 1/(d-p+1) \text{ otherwise } 0$

Change the business role BookingAgent to let customer pay CR.BOOKFEE to the payagent (ignore the problem of “distributing” the amount between TravelBooking and FlightAgent)

Change the business protocol Customer to ensure that the parameter amount of refund is equal to the amount of the trip minus the booking fee (which is not refunded)

BUSINESS ROLE BookingAgent is

INTERACTIONS

...

```
r&s bookTrip
  ⚠ from,to:airport
  out,in:date
  ☒ fconf:fcode
  hconf:hcode
  amount:moneyvalue
ask log(username,password):bool
ask getData(username):usrdata
ask getCard(username):paydata
s&r bookFlight
  ⚠ from,to:airport
  out,in:date
  traveller:usrdata
  ☒ fconf:fcode
  amount:moneyvalue
  beneficiary:accountn
  payService:serviceId
s&r payment
  ⚠ amount:moneyvalue
  beneficiary:accountn
  originator:usrdata
  cardNo:paydata
  ☒ proof:pcode
s&r bookHotel
  ⚠ checkin,checkout:date
  traveller:usrdata
  ☒ hconf:hcode
snd payAck
  ⚠ proof:pcode
  status:bool
rcv ackRefundRcv
  ⚠ amount:moneyvalue
snd ackRefundSnd
  ⚠ amount:moneyvalue
```

SLA VARIABLES

BOOKFEE: [0..100], KD: [1..30]

ORCHESTRATION

```
local s:[START, LOGGED, QUERIED, FLIGHT_OK, HOTEL_OK, CONFIRMED, END_PAID,
END_UNBOOKED, COMPENSATING, END_COMPENSATED],
logged:bool, traveller:usrdata, travcard:paydata
```

...

transition HotelAnswer

```
triggeredBy bookHotel☒
guardedBy s=FLIGHT_OK
effects bookHotel.Reply ⊃ s'=HOTEL_OK
  ∧ ¬bookHotel.Reply ⊃ s'=END_UNBOOKED
sends bookHotel.Reply ⊃ bookTrip☒
  ∧ bookTrip.fconf=bookFlight.fconf
  ∧ bookTrip.amount=bookFlight.amount + BOOKFEE
  ∧ bookTrip.hconf=bookHotel.hconf
  ∧ ¬bookHotel.Reply ⊃ bookFlight*
  ∧ bookTrip☒
  ∧ bookTrip.Reply=False
```

transition TripCommit

```
triggeredBy bookTrip✓
guardedBy s=HOTEL OK
effects s'=CONFIRMED
sends bookFlight✓ ∧ bookHotel✓ ∧ payment△
  ∧ payment.amount=bookFlight.amount + BOOKFEE
  ∧ payment.beneficiary=bookFlight.beneficiary
  ∧ payment.originator=traveller
  ∧ payment.cardNo=travcard
```

transition PaymentAnswer

```
triggeredBy payment☒
guardedBy s=CONFIRMED
effects payment.Reply ⊃ s'=END_PAID
  ∧ ¬payment.Reply ⊃ s'=END_UNBOOKED
sends payAck△
  ∧ payAck.proof=payment.proof
  ∧ payAck.status=payment.Reply
```

...

BUSINESS PROTOCOL Customer is**INTERACTIONS**

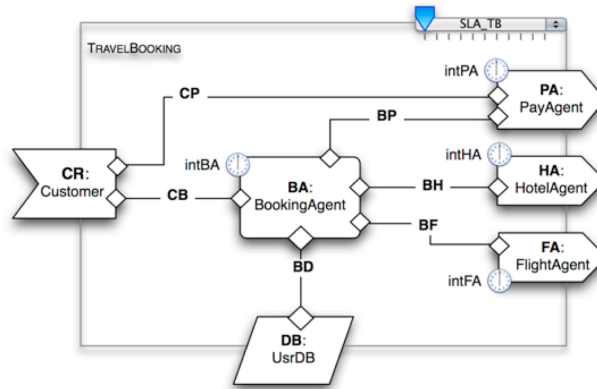
```
r&s login
  △ usr:username, pwd:password
r&s bookTrip
  △ from,to:airport,
  out,in:date
  ☒ fconf:fcode,
  hconf:hcode,
  amount:moneyvalue
snd payNotify
  △ status:bool
snd refund
  △ amount:moneyvalue
```

SLA VARIABLES

```
BOOKFEE:[0..100], KD:[1..30]
```

BEHAVIOUR

```
initiallyEnabled login△?
(login☒! ∧ login.Reply) enables bookTrip△?
(bookTrip☒ ∧ bookTrip✓?) ensures payNotify△!
(payNotify△! ∧ payNotify.status) enables bookTrip†?
  until today≥bookTrip.out+KD
(bookTrip†? ∧ today+KD<bookTrip.out) ensures refund△!
  ∧ refund.amount=bookTrip.amount-BOOKFEE
```



Define, for the module *TravelBooking*, an external policy on the SLA variables:

- *CR.PROCODE*: is a variable associated to *CR* that denote the promotional code used by the customer to obtain discounts,
- *CR.PERC*, *FA.PERC*: is a variable associated to *CR* that denote the percentage of refund,
- *CR.FEE*: is the variable associated to *CR* that denote the forfait price for each booking,

The external policy must contain a number of constraints that ensure:

- That the percentage of refund conceded to the customer must always be between 50% and 100% and the flight agent must support this.
- The degree of satisfaction is inversely proportional to the percentage of refund (PERC) conceded to the customer.
- If the promotional code of the customer is “VIP” then *CR.FEE* is the zero, if it is “MEMBER” the satisfaction is directly proportional to *CR.FEE* but less than 10£ and if it is “OTHER” then it 11£.

EXTERNAL POLICY

SLA VARIABLES

CA.PERC, FA.PERC, CR.FEE, CR.PROCODE

CONSTRAINTS

C₁: {CA.PERC, FA.PERC}

$$def(x,y) = \begin{cases} 1 & \text{if } 50 \leq x \leq 100 \wedge x \leq y \\ 0 & \text{otherwise} \end{cases}$$

C₂: {CA.PERC}

$$def(s) = \begin{cases} 1/x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

C₃: {CR.FEE, CR.PROCODE},

$$def(f,p) = \begin{cases} 1 & \text{if } (p = \text{"VIP"} \wedge f = 0) \vee (p = \text{"OTHER"} \wedge f = 11) \\ f/10 & \text{if } p = \text{"MEMBER"} \text{ and } f \leq 10 \\ 0 & \text{otherwise} \end{cases}$$

BUSINESS PROTOCOL Customer is

INTERACTIONS

```
r&s login
  Ⓐ usr:username, pwd:password
r&s bookTrip
  Ⓐ from,to:airport,
    out,in:date
  ☒ fconf:fcode,
    hconf:hcode,
    amount:moneyvalue
snd payNotify
  Ⓐ status:bool
snd refund
  Ⓐ amount:moneyvalue
```

BEHAVIOUR

```
initiallyEnabled loginⒶ?
(login☒! ^ login.Reply) enables bookTripⒶ?
(bookTrip☒ ^ bookTrip✓?) ensures payNotifyⒶ!
...
```

Define the following statement for the business protocol Customer:

- That the percentage of refund (*Refund.amount*) conceded to the customer must always be between 50% and 100%.

```
50 ≤ refund.amount ^ refund.amount ≤ 100 after refundⒶ!
```

- That the compensation of *bookTrip* is always allowed after a *payNotify* with a positive status (one statement) but that the amount refund (which is ensured after *payNotify*) will be zero on or after the day of the trip (one statement).

```
1- payNotifyⒶ! ^ payNotify.status enables bookTrip‡?
```

```
2- Refund.amount=0 after bookTrip.out ≥ today
```

```
(bookTrip‡? ensures refundⒶ!) this was assumed...
```

- That (alternative to the previous) the compensation of *bookTrip* is not allowed on or after the day of the trip (one statement) but the amount of the refund is always as the one agreed with the SLA variable *PERC* (one statement).

1- `payNotify! ^ payNotify.status enables bookTrip? until bookTrip.out ≥ today`

2- `refund.amount=bookTrip.amount*PERC after refund!`

(we should declare the SLA variable *PERC* in *Customer*)