SRML Editor Tutorial (3)

CO7205 Advanced System Design

José Fiadeiro
Laura Bocchi
Yi Hong

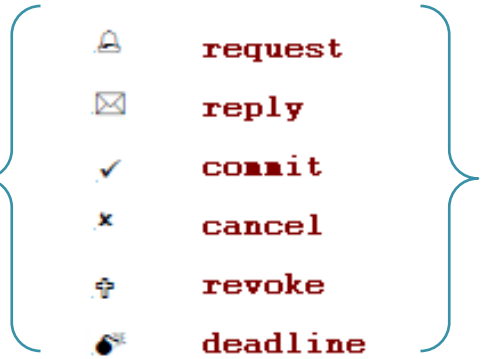
Lab sheet and materials

http://www.cs.le.ac.uk/srml/

Business Role and Orchestration

SRML Notation	Editor Text
s’	*post(s)*
^	*&*
v	|
⊃	*implies*
bookTrip	*bookTrip[request]*
¬ true	*!true*

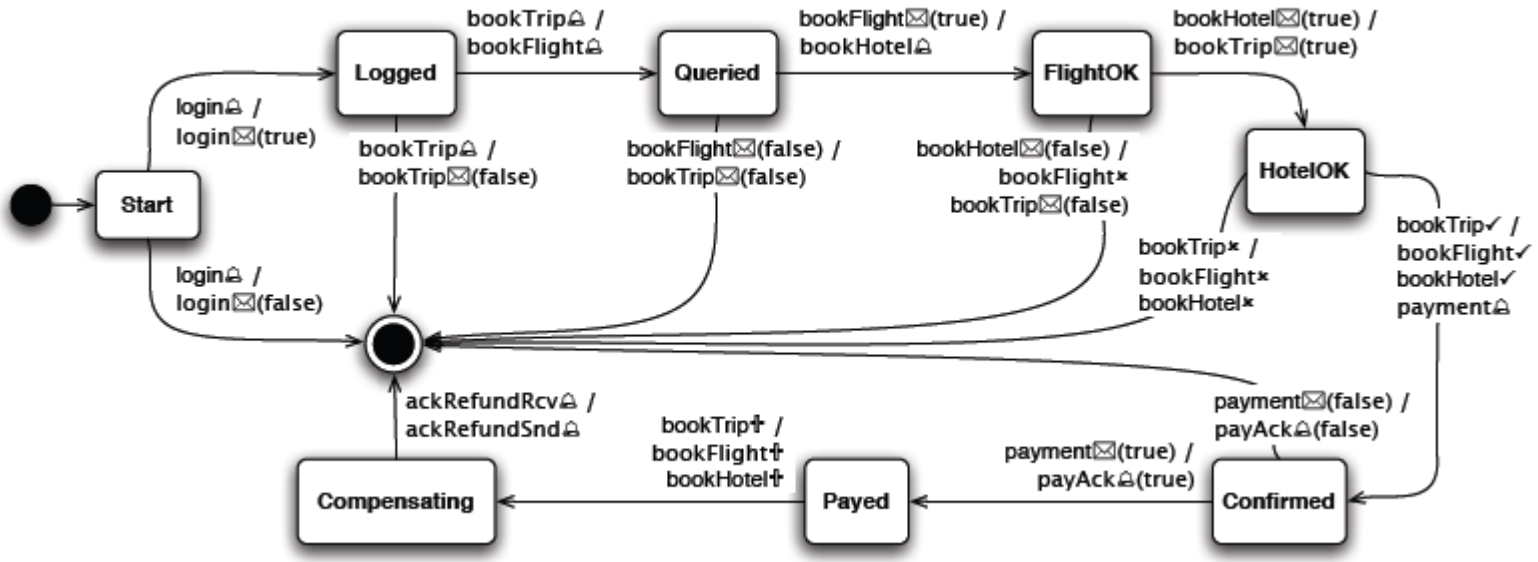
Corresponding textual representation of SRML notation



* interaction event type ‘deadline’ is only available from version 1.5.0,

SRML Orchestration

Define state variables used for state machine:



SRML Orchestration

Steps (1)

declare an enumeration type (e.g. *state***)** in the data type section.

```
datatype TravelBookingTypes is
    imports BOOL;
    imports STRING;
    ...........
    sort usrname mappedTo String;
    sort password mappedTo String;
    ...........
    enum state { START, LOGGED, QUERIED, FLIGHT_OK, HOTEL_OK,
                 END_PAYED, END_UNBOOKED, COMPENSATING, END_COMPENSATED };
endd
```

Steps (2)

declare a local variable (e.g. *s*) of type *state* in the orchestration section

```
orchestration
  local  s:state, logged:bool, traveller:usrdata
```

Business Role and Orchestration

Syntax of orchestration:

orchestration

local *local_variable_name* : *data_type* **, …..**

transition *transition_name_1* **{**
triggeredBy …
guardedBy …
effect …
sends …
}

transition *transition_name_2* **{**
….
}

Local variables can be:
(1) State variables
(2) Other variables

data_type must be declared.

A *trigger* is either the process of an event or a state condition

A *guard* is a condition that identified the states in which the transition can occurs

effect specifies the effects of the local state

sends specifies the events that are published during the transition.

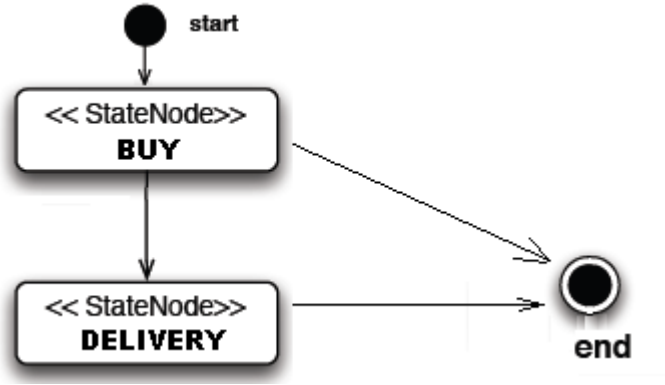
* BNF-liked grammar for orchestration:
http://www.cs.le.ac.uk/srml/example/srml_grammar.txt

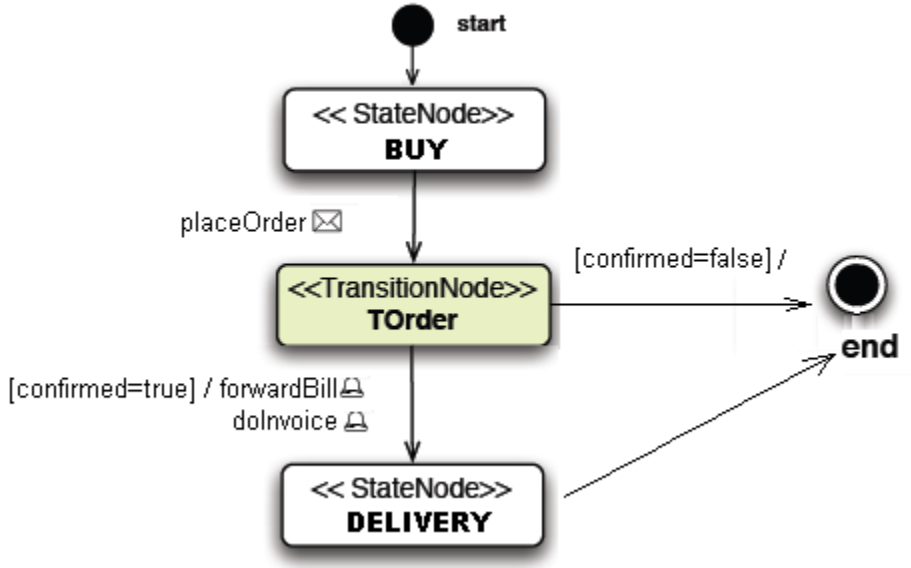
Business Role and Orchestration

- Example:

  orchestration with one transition and two state nodes:

(1) Identify state and transition node
(2) event/condition/action

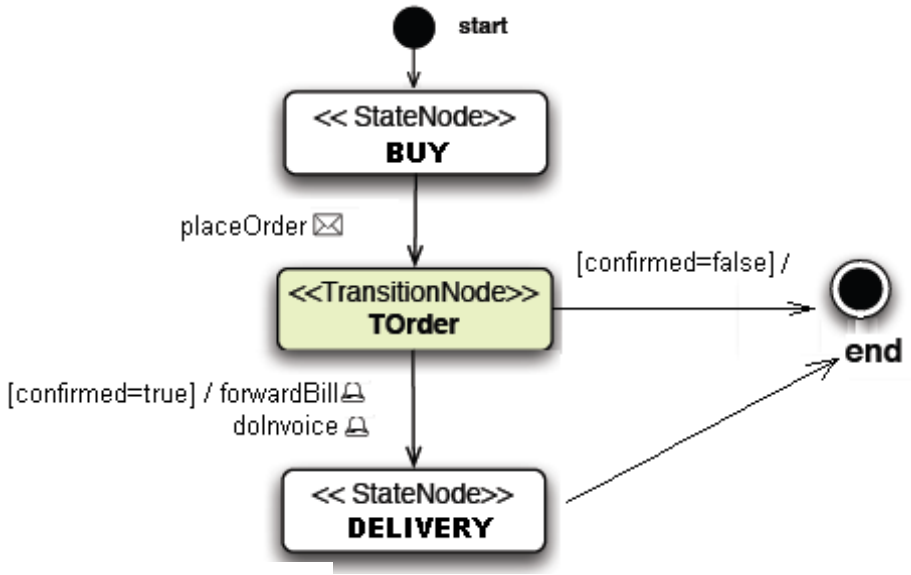




Business Role and Orchestration

An example of orchestration with one transitions:

```
datatype BOOL is
    sort bool mappedTo Bool;
endd

datatype EasyBankType is
    sort ProductType;
    sort BillType mappedTo Int;
    sort usrname mappedTo String;
    sort password mappedTo String;
    sort DBInfo mappedTo String;

    enum state { START, BUY, DELIVERY, END};

endd

.........

orchestration

 local s:state, confirmed:bool


 transition TOrder{
     triggeredBy placeOrder[request ]
     guardedBy s = BUY
     effects
         post(confirmed)=authUser(authUser.name, authUser.pwd)
             & post(confirmed) implies post(s) = DELIVERY
             &!post(confirmed) implies post(s) =END
     sends placeOrder[reply] & placeOrder.Reply=post(confirmed)
         & post(s)= DELIVERY implies
             doinvoice[request] & doinvoice.product=placeOrder.product &
             forwardBill[request] & forwardBill.product=placeOrder.product
 }
```



Practice

Review Lecture Note:

https://campus.cs.le.ac.uk/teaching/resources/CO7205/ASD.11.pdf

Now try the following:

(1) Create orchestration section within Business Role using source code view.
(2) Declare local variables.
(3) Identify transition and state nodes from state machine diagram.
(4) define state enumeration data type for states (or using SRML wizard).
(4) Identify trigger, guard, effects and send
(5) Enter appropriate textual representation of SRML notations