INF5150 Suggested solution to the first obligatory exercise

October 20, 2008

Solve Exercise 2 on STAIRS in the exam paper for 2007. The sequence diagrams are on the last pages.

2.

(a) *Events*

The STAIRS Tutorial associates two events with each message, a transmission event and a reception event.

    i. Q: What is (are) the first event(s) of Figure 2. (If there is more than one possibility, list all the possibilities.)

A: There is only one first event: !(*Call-for-proposals*, *Peter*, *syst:System*).

Explanation:

- Weak sequencing requires that events from the same lifeline are ordered in the trace in the same order as on the lifeline. Hence, we only consider events at the top of each lifeline.

- The principle of causality requires that a message can never be received before it has been transmitted. The transmission event for a message is therefore always ordered before the reception event for the same message. Peter is the only lifeline that begins with a transmission event. Therefore this is the only possible first event.

    ii. Q: What is (are) the last event(s)?

A: {?(*Reject*, *syst:System*, *Paul*), ?(*Evaluation*, *Peter*, *syst:System*)}

Explanation:

- Since *Negotiation* ends with an **alt**-fragment, the last events are the ones from the union of the set of traces obtained when sequencing the fragment before the **alt** with the set of traces from each operand.

- The only last event when sequencing the upper fragment with the first operand is ?(*Reject*, *syst:System*, *Paul*), since Paul is the only lifeline ending with a receive event here.

- The only last event when sequencing the upper fragment with the second operand is ?(*Evaluation, Peter, syst:System*), since syst:System is the only lifeline ending with a receive event here.

(b) *Traces*

    i. Q: How many traces are there in the **alt**-construct (combined fragment) inside *Negotiation* (Figure 2.)
A: There are seven traces.

Let

peres = (*Reject, Peter, syst:System*)   srepa = (*Reject, syst:System, Paul*)
peas = (*Accept, Peter, syst:System*)   sapa = (*Accept, syst:System, Paul*)
pees = (*Evaluation, Peter, syst:System*) paes = (*Evaluation, Paul, syst:System*)

The first operand has exactly one trace: $\langle !peres, ?peres, !srepa, ?srepa \rangle$

The second operand has six traces, due to the fact that sending of Evaluation from Peter is independent of all events on the other lifelines except reception of that message, see Figure 1.
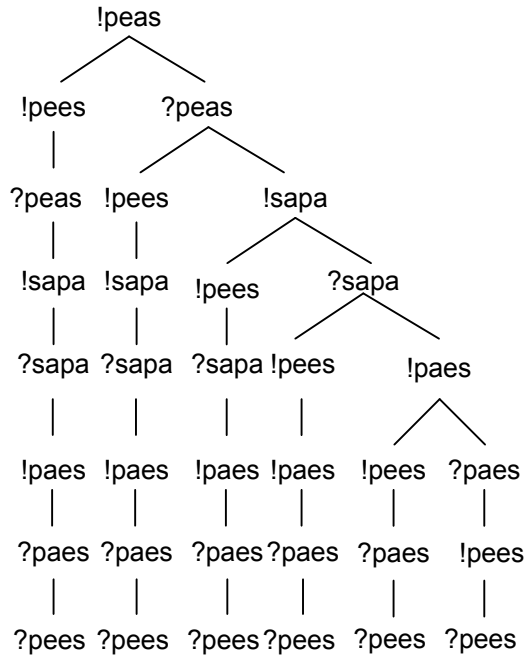

Figure 1: The traces of the second **alt**-operand

    ii. Q: How many traces are negative with respect to *Negotiation* (Figure 2.)
A: None.
Explanation: The sequence diagram uses no constructs for specifying negative behaviour, such as refuse, veto, assert or guards.

(a) *Refinement*

    i. Q: Assume that Bob has joined the meeting group. We augment our *Negotiation* with one more lifeline representing Bob. He behaves like Mary and refuses the meeting call-for-proposals e.g. as represented in Figure 3. Explain whether the augmented *Negotiation2* is a supplementing of the original *Negotiation* (Figure 2) or not.

       A: The augmented diagram *Negotiation2* is not a supplementing of the *Negotiation*.

       Explanation: None of the positive traces in the single interaction obligation representing the original diagram, are included in the interaction obligation representing the augmented diagram.

   ii. Q: It turns out that Peter is a very easy-going fellow that would not dream of rejecting meeting proposals. How would you modify the *Negotiation* (Figure 2) to accommodate for Peter's attitude such that the modified behavior is a narrowing of the original?

       A: One possible solution is to put the *Reject* messages of the first **alt**-operand within a **refuse**-construct. Thus all traces ending in a rejection are moved from the set of positive traces to the set of negative traces.
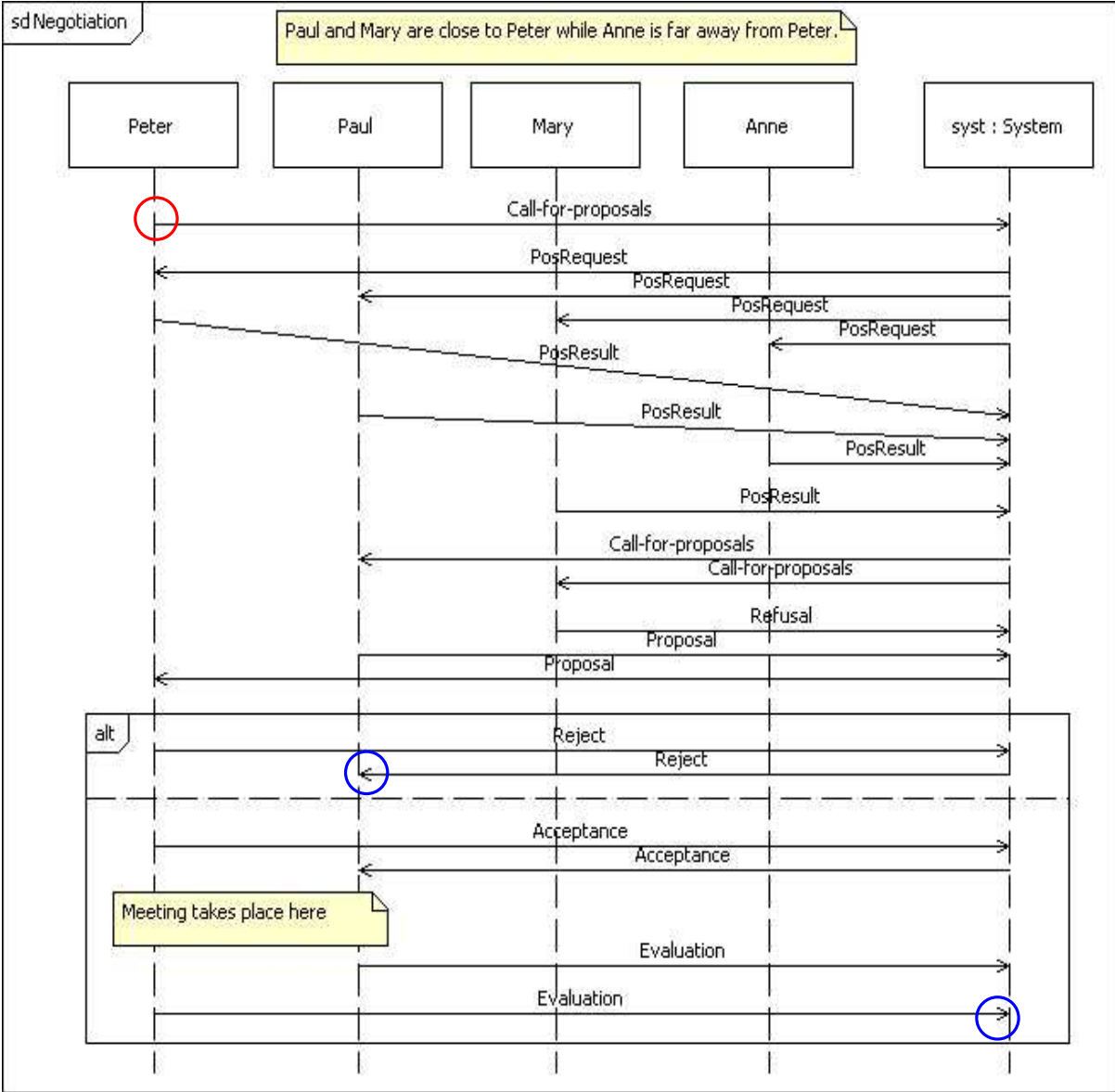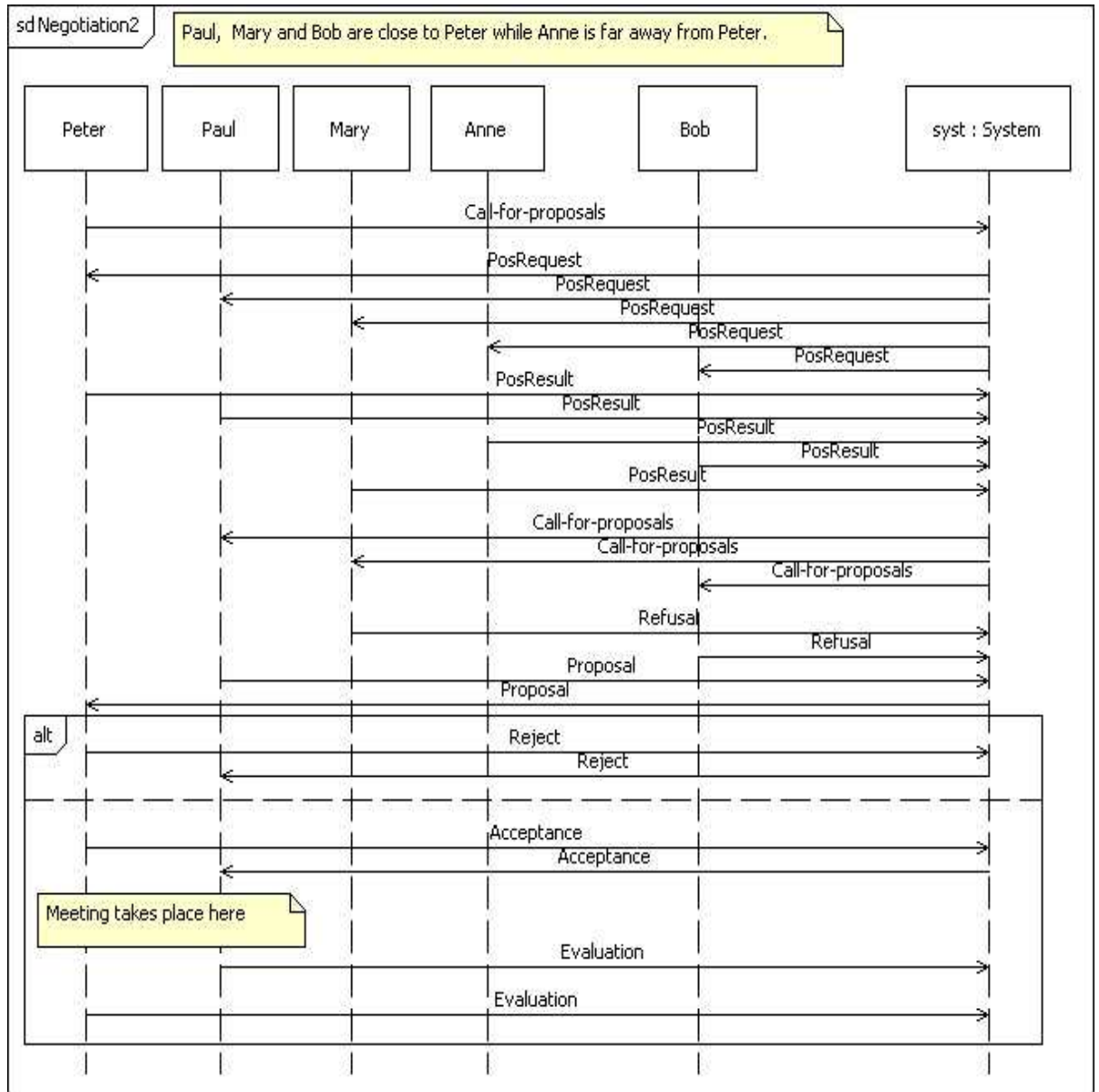
Figure 2: The negotiation for a spontaneous meeting

Figure 3: Augmented *Negotiation2* where Bob has been added