

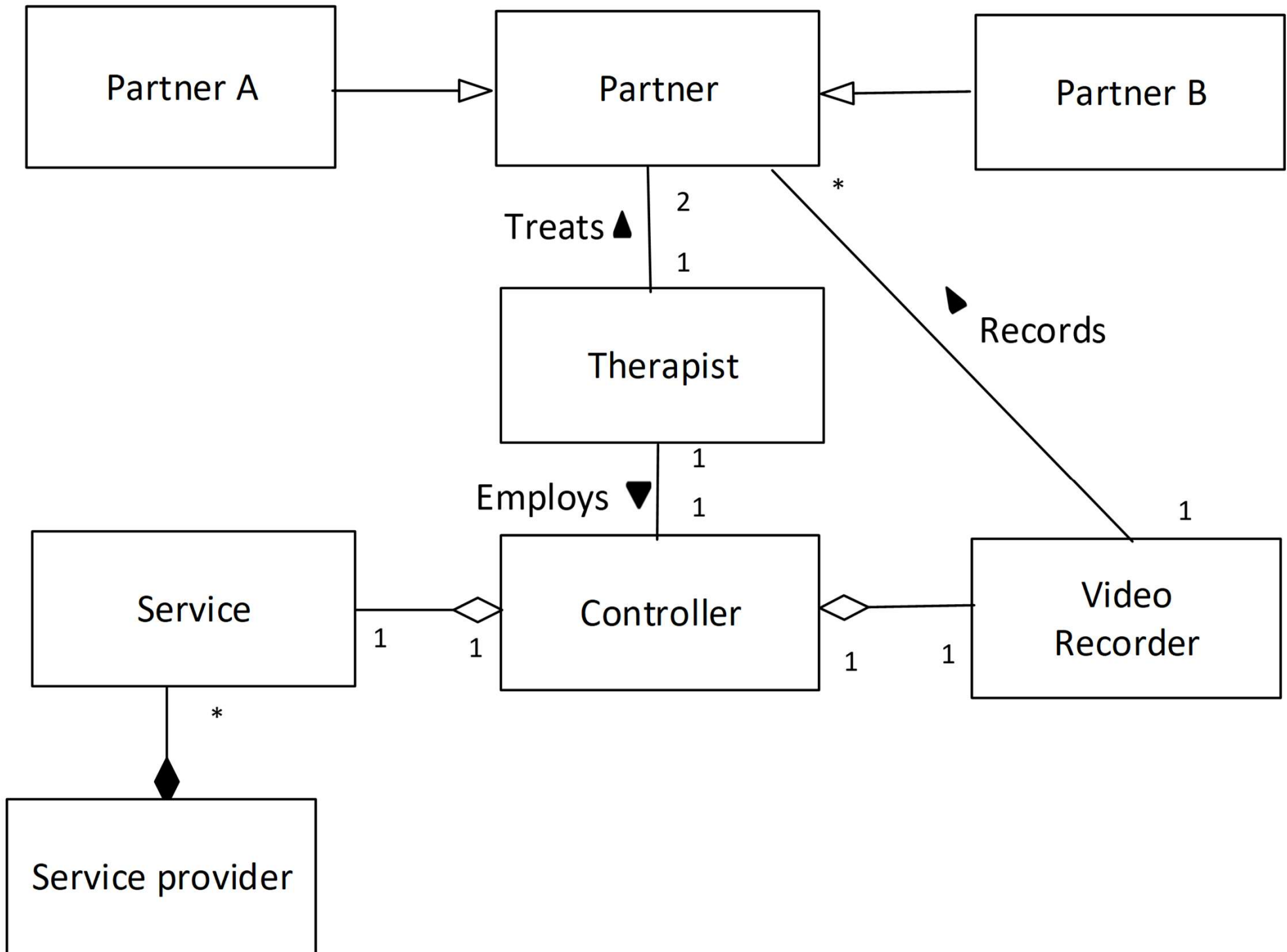
Solution proposal

Oblig-I Modeling

Ketil Stølen

Task I

- Make a class-diagram describing the relevant concepts. The class-diagram should at least represent the following concepts: service, video recorder, therapist, partner-a, partner-b, controller (used amongst others to steer the recorder from one partner to the other). The class-diagram should contain the generalization relation and illustrate correct use of both composition and aggregation. All associations should be decorated with multiplicities.



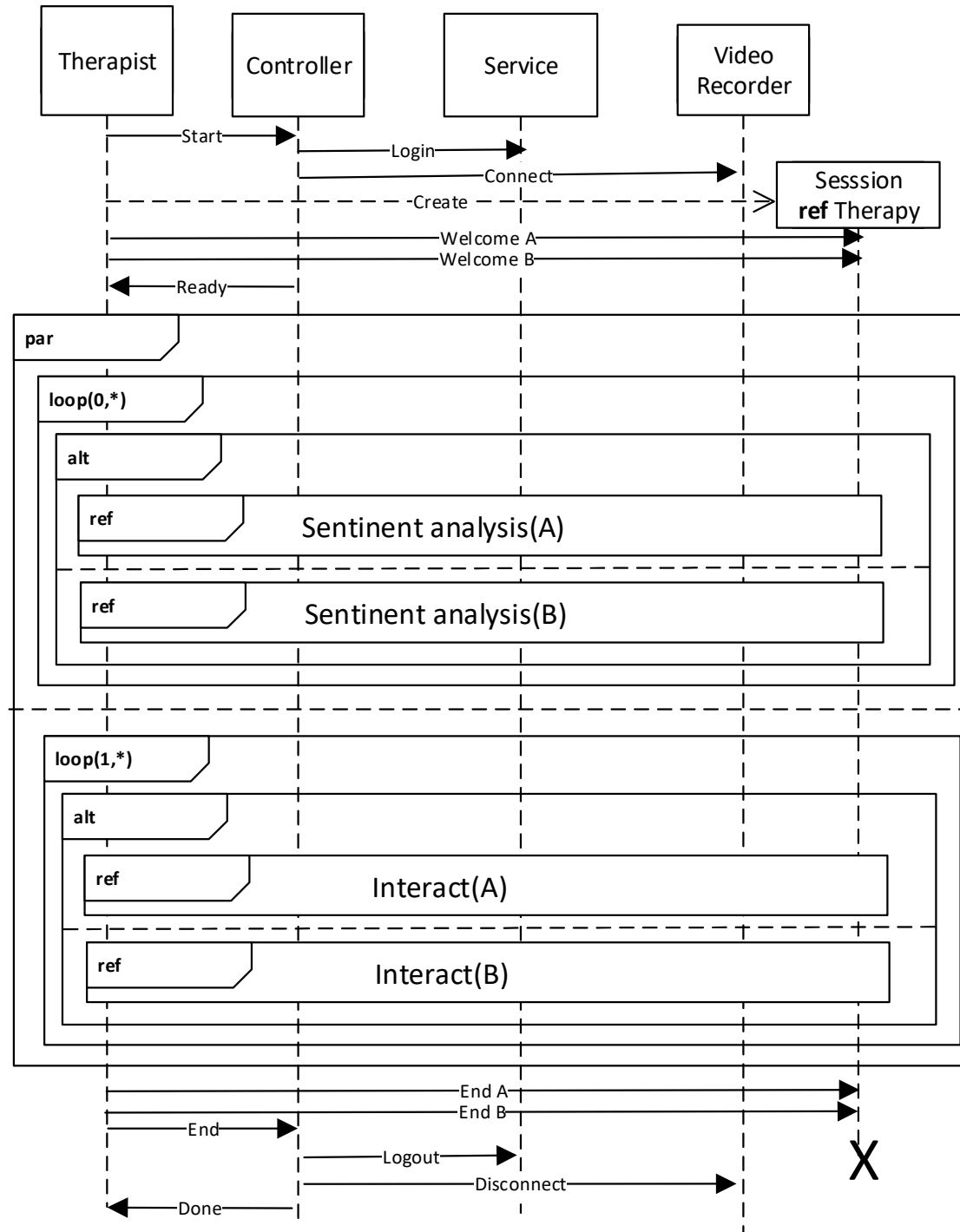
Remember

- Composition has by default multiplicity 1 at the diamond end
- The same class cannot be associated with two different "diamond-end" classes
- Class names should always be in singular

Task II

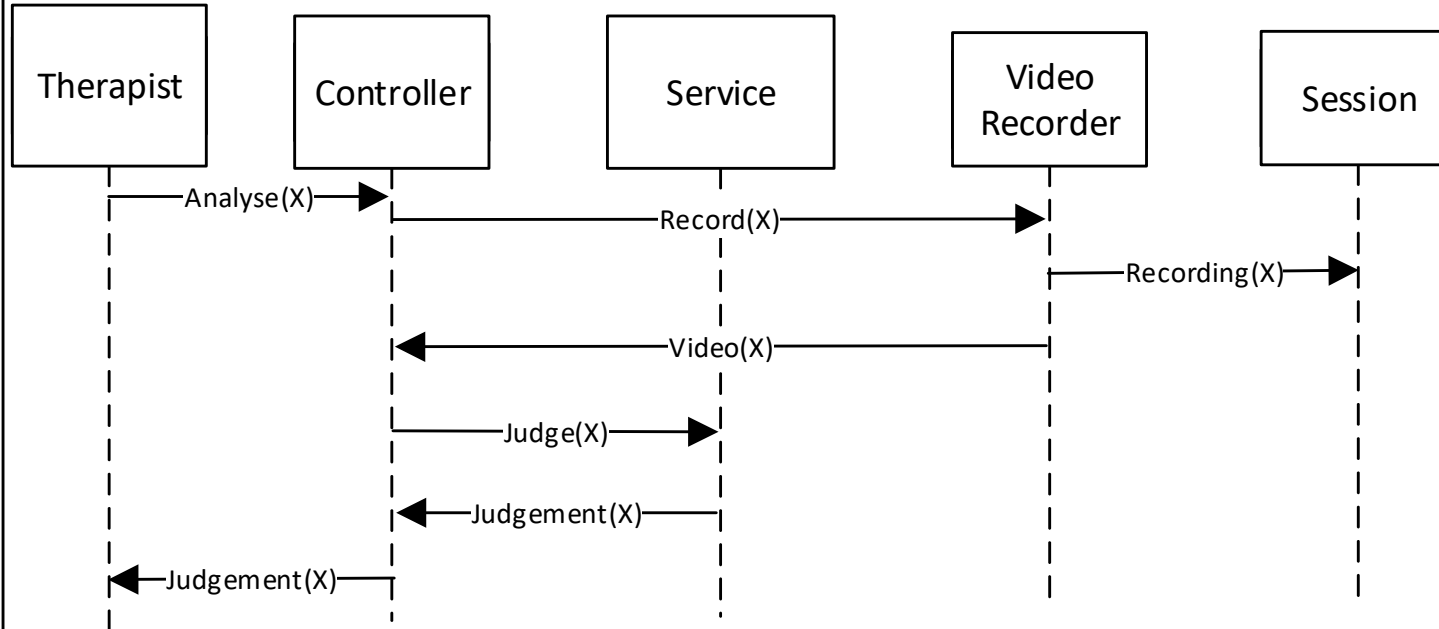
- Make one main sequence diagram describing a therapy session. There should at least be lifelines for therapist, service, controller, video recorder, partner-a and partner-b. The sequence diagram should illustrate correct use of at least referencing (the referenced sequence diagrams should also be specified in full), loop, lifeline creation and termination, and opt. The sequence diagram should also describe some negative behaviour. The sequence diagram should provide a basic representation of the interactions between the involved actors during a therapy session. Please avoid excessive complexity. The sequence diagram being made in this task will be used further in the next task.
- Describe the first ten events of at least one positive trace represented by your sequence diagram.

sd Therapy Session



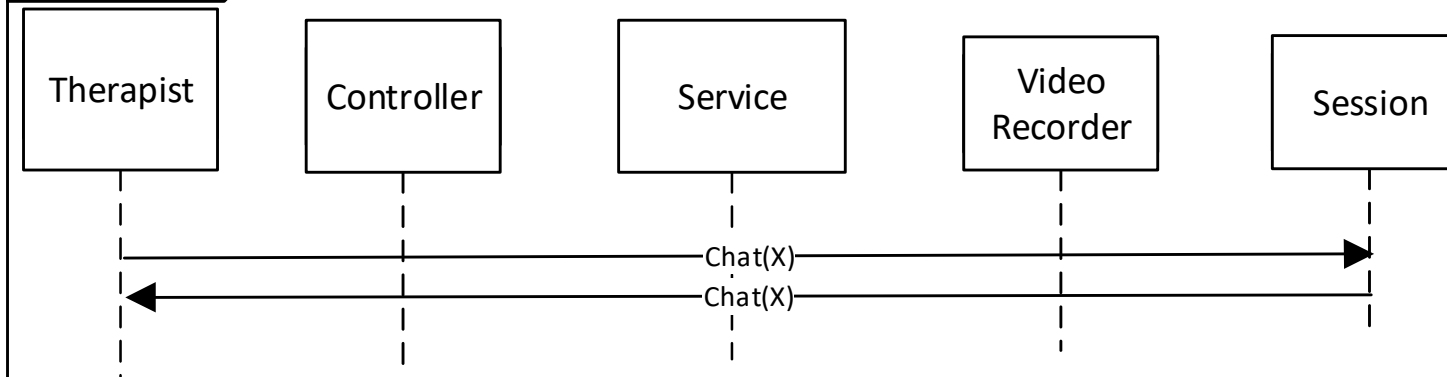
sd Sentinent Analysis(X)

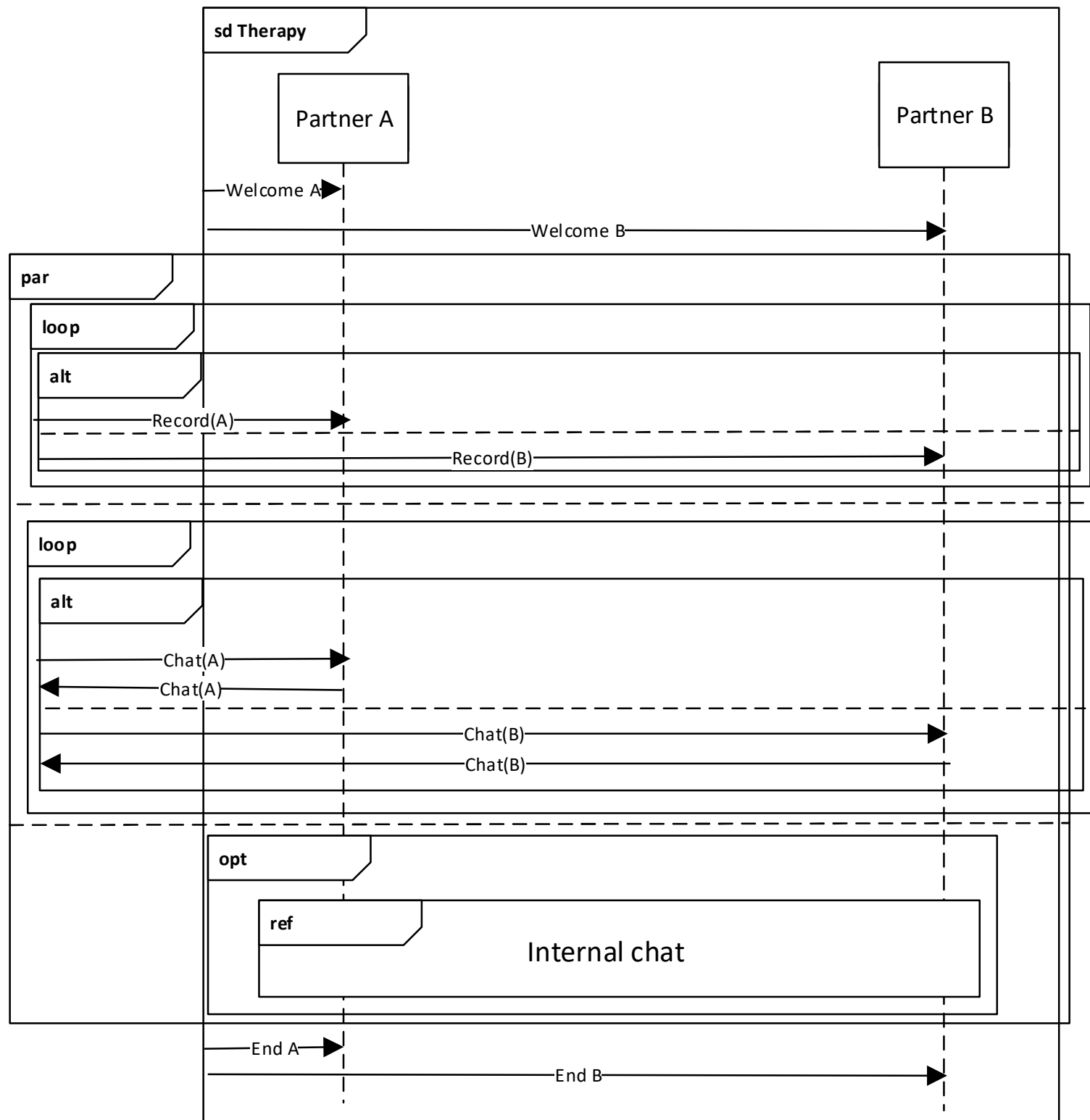
assert



sd Interact(X)

assert





Note the way
par, loop and alt
are outside the
main frame to
specify that
these are global
to Therapy
(matching the
structure of
Therapy Session)

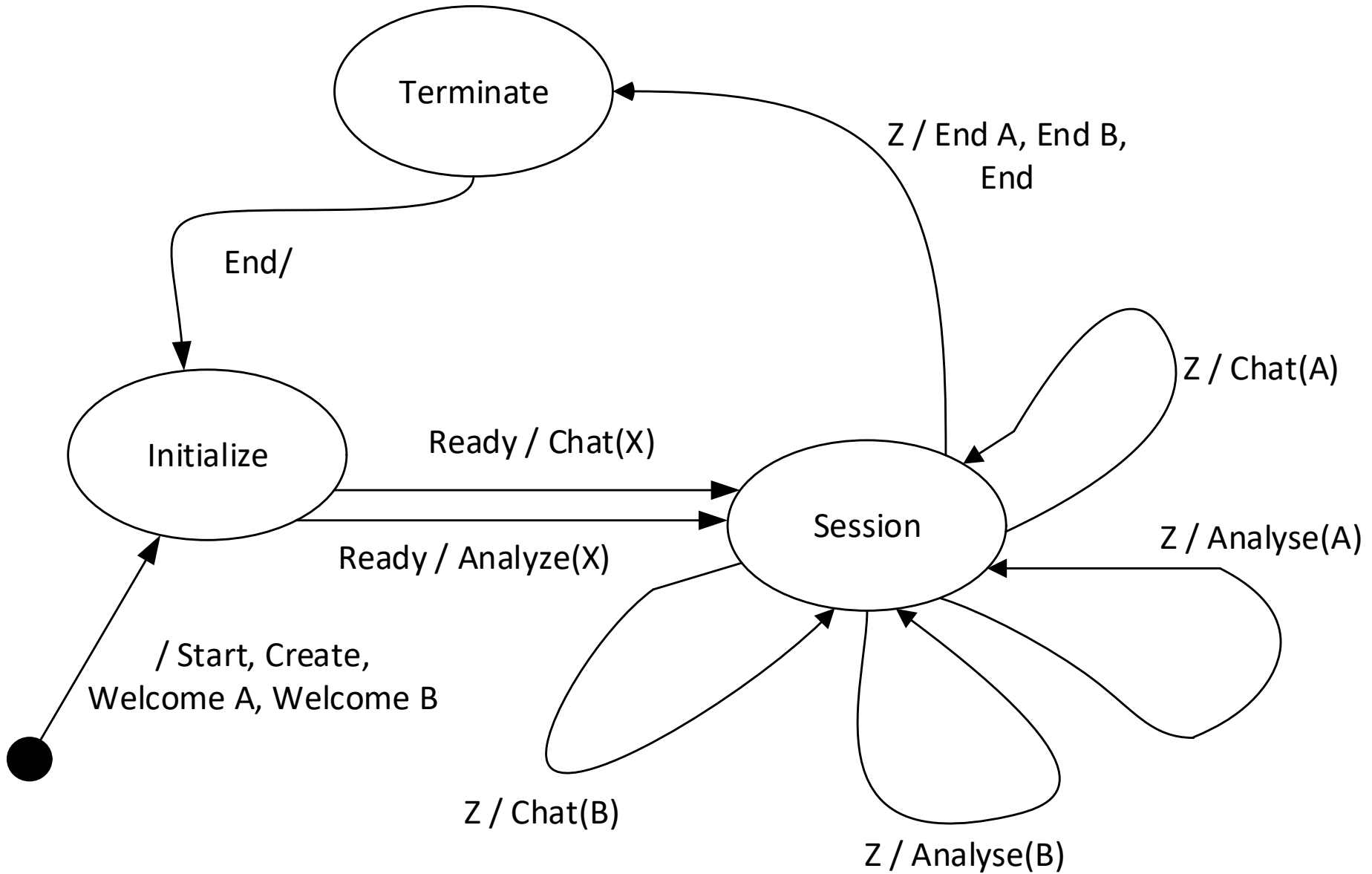
First ten events of a trace

<!start, ?start, !login, ?login, !connect, ?connect, !create, ?create,
!welcome A, ?welcome A, ... >

Task III

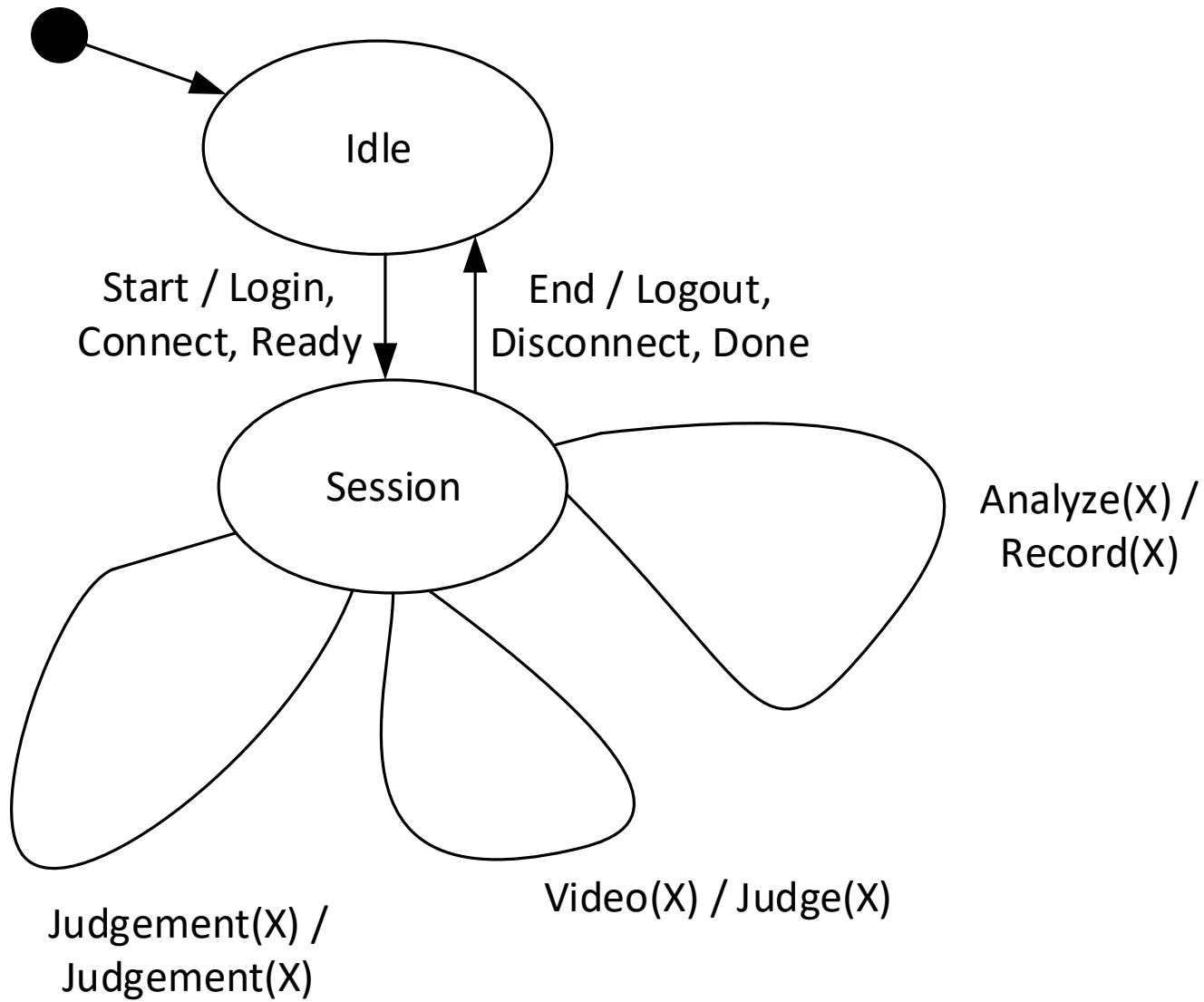
- Make a communicating state-machine that correctly reflects all the traces specified by the sequence diagram for the **controller lifeline**.
- Make a communicating state-machine that correctly reflects all the traces specified by the sequence diagrams for the **therapist lifeline**.
- The two state-machines must allow any positive trace allowed by the respective lifelines in the sequence diagrams but may have additional traces.
- Remember that the machines should **communicate the messages specified** by the sequence diagrams. **Hence, for a lifeline, an input signal (output signal) in a state machine should be the reception (sending) of a message in the sequence diagram.**

sm Therapist



Z is either Chat(A), Chat(B), Judgement(A) or Judgement(B)

sm Controller



Remember

- Any positive behaviour of the relevant lifeline should be allowed
- Any negative behaviour of the relevant lifeline should not be allowed
- The signals in the state machines should be messages in the sequence diagrams
- The state machines must be reactive
- The initial transition in a state machine may send signals