SINTEF

70 years
1950-2020

PROPOSED SOLUTION OBLIG-I 2020 WITH ADDITIONAL FEEDBACK

In general

- Those who have got their oblig accepted will receive an email with some sentences with feedback.
- This feedback will not be detailed because most weaknesses are quite common.
- I will instead try to address these weaknesses in a careful manner in this presentation.
- In addition I will present a full solution proposal.

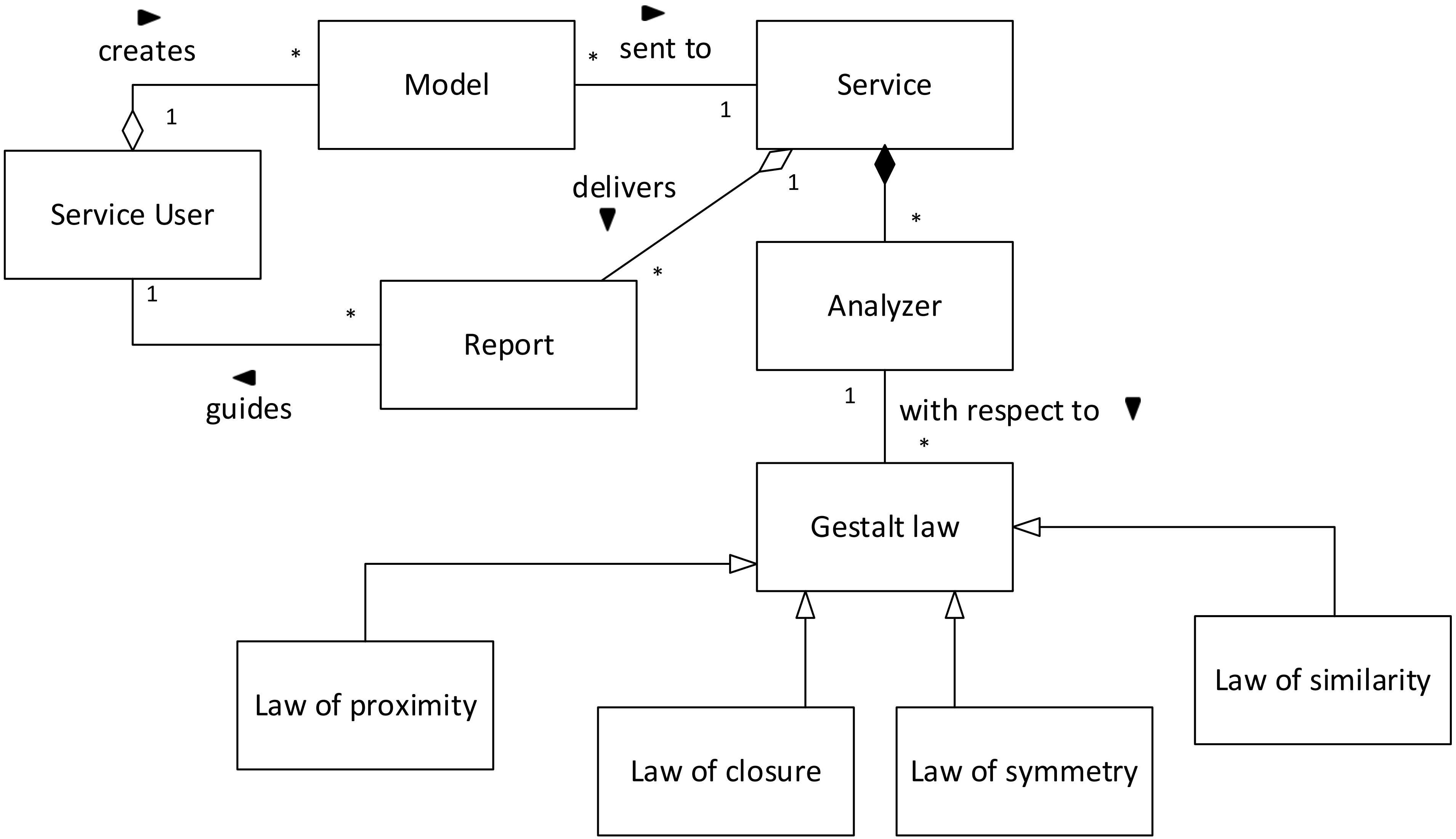
SINTEF

70 år
1950-2020

Task I

General feedback on submitted obligs

- Most solved this task quite ok
- Make sure that composition (black diamond) is used correctly
  - Always wrong if the same (child) class is composed into two different (parent) classes
  - If the parent class is deleted so is the child classes. *Composition* implies a relationship where the child cannot exist independent of the parent.
- No multiplicity on an association means any number (except for parent of composition for which 1 is default)
- Use arrows on association predicates to specify reading direction


creates
*
1
Model
*
sent to
1
Service
Service User
delivers
1
*
*
Analyzer
1
*
Report
guides
1
with respect to
*
Gestalt law
Law of proximity
Law of closure
Law of symmetry
Law of similarity

SINTEF

70 år

1950-2020

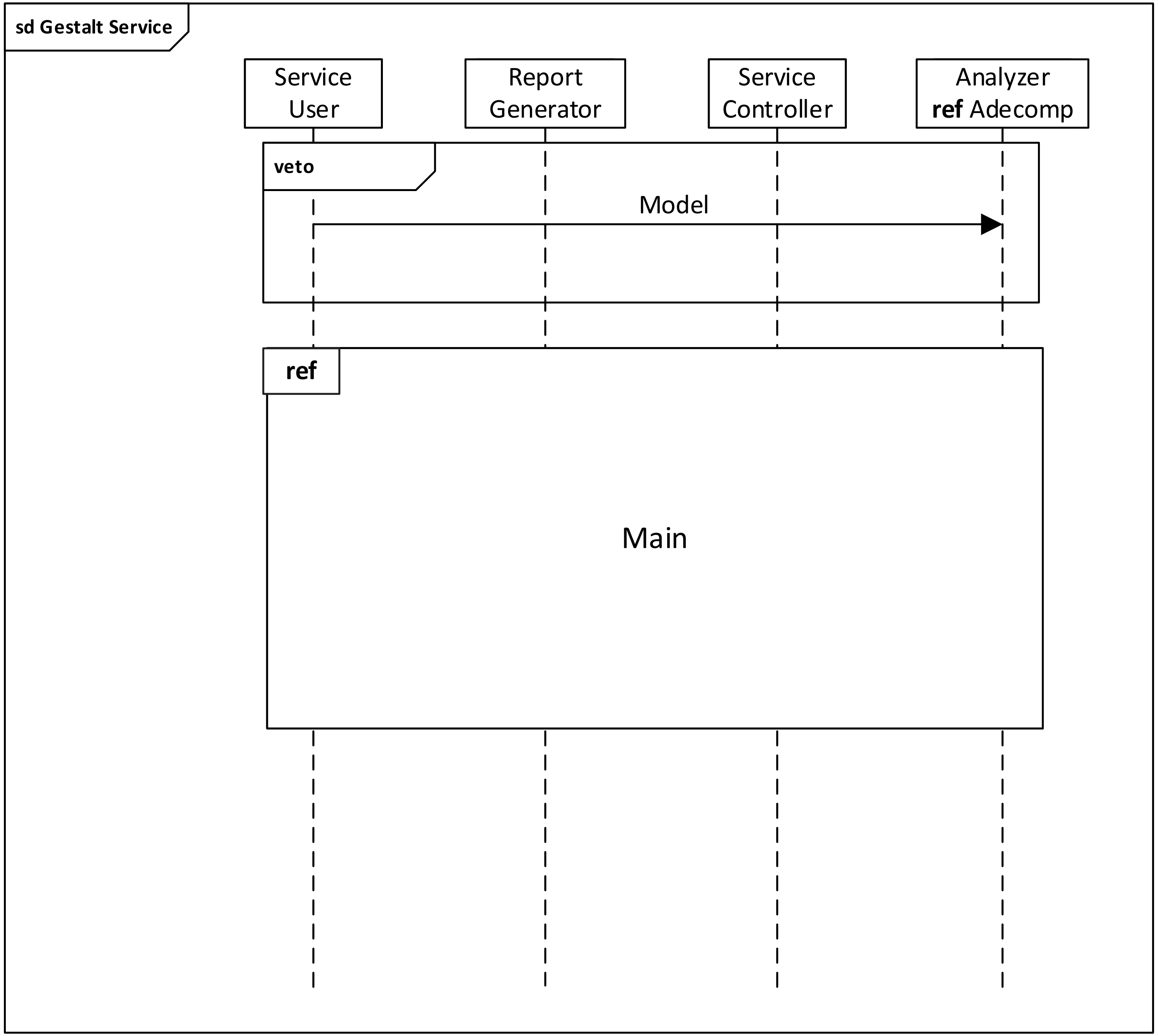
Task II

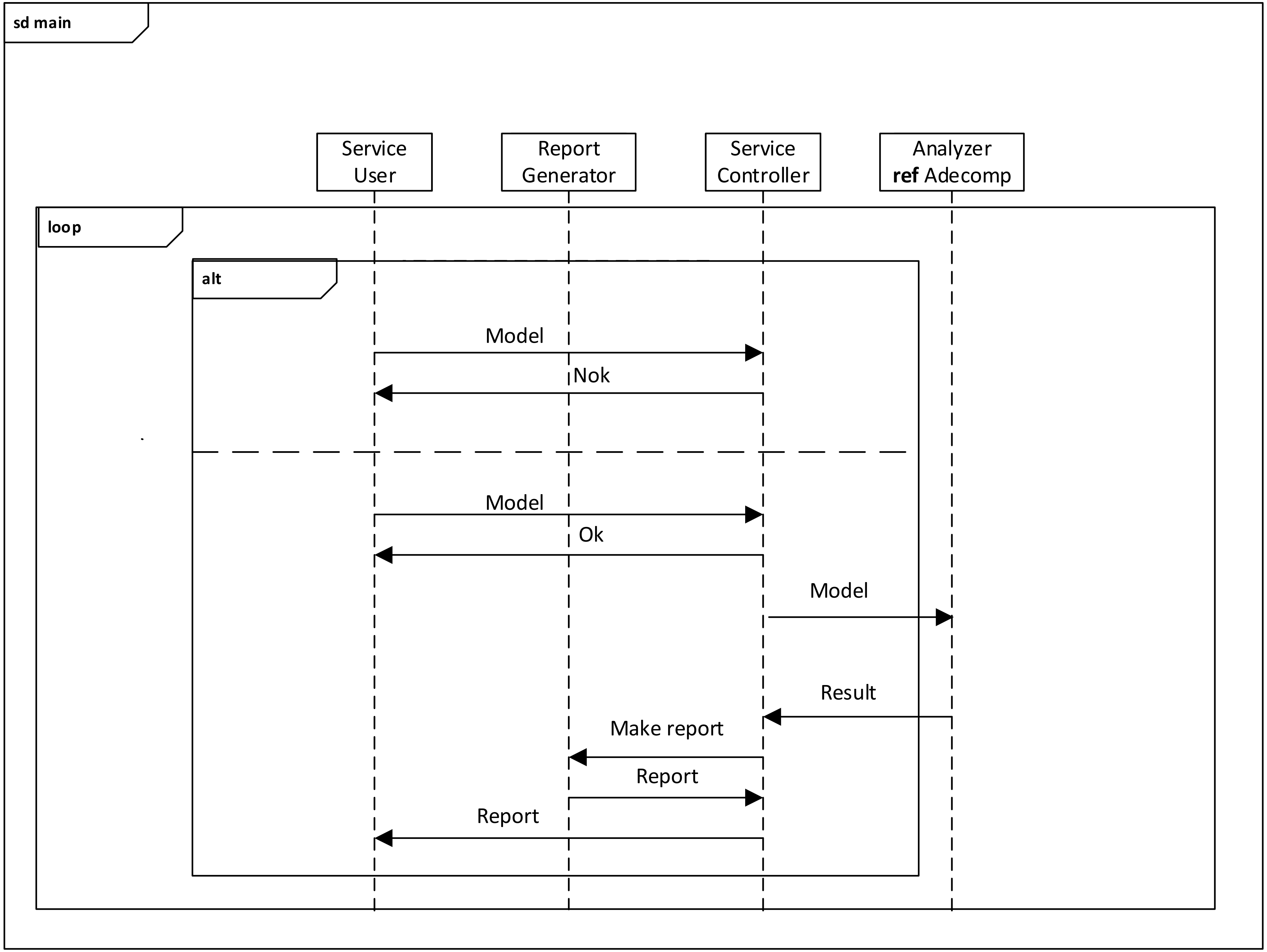
General feedback on submitted obligs

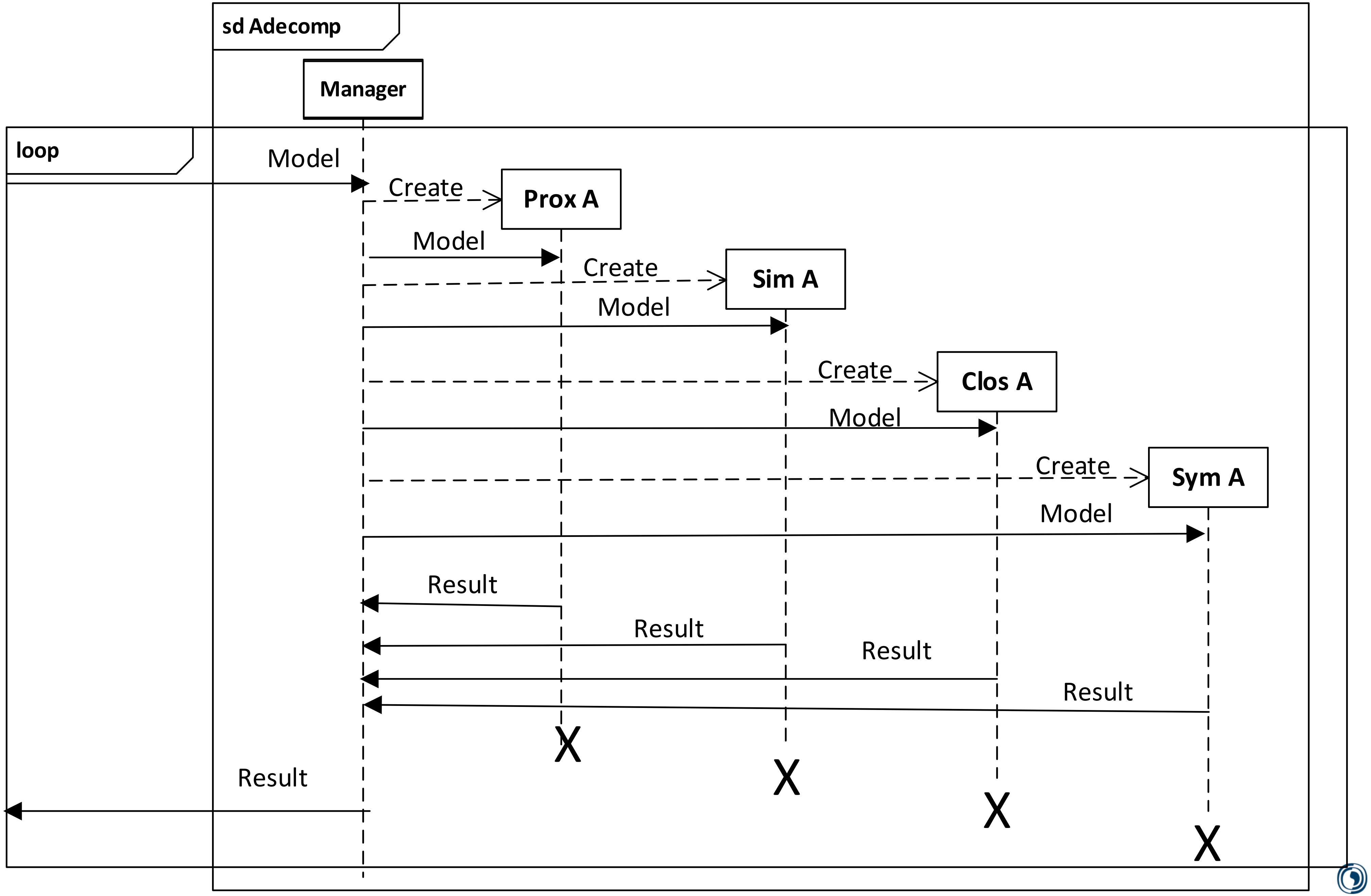
- The lifeline decomposition concerns a single lifeline
- When a lifeline is decomposed then this lifeline is represented by a set of brand-new lifelines in the diagram resulting from the decomposition
- Make sure that the external interaction in the new diagram match 100% the communication history of the lifeline being decomposed
  - The message names should be the same
- The new diagram may contain arbitrarily many new internal interactions

General feedback on submitted obligs

- Make sure that global operators involving the lifeline being detailed are global in the detailed diagram (see Adecomp below)
- Negative behaviour is behaviour that should not be produced by the specified service
- Many of you specify error handling as negative. This does not make sense.
- An **alt** within a **loop** may choose the very same alternative in every iteration


sd Gestalt Service
Service User
Report Generator
Service Controller
Analyzer **ref** Adecomp
veto
Model
ref
Main


sd main
Service User
Report Generator
Service Controller
Analyzer
ref Adecomp
loop
alt
Model
Nok
Model
Ok
Model
Result
Make report
Report
Report


sd Adecomp
Manager
loop
Model
Create
Prox A
Model
Create
Sim A
Model
Create
Clos A
Model
Create
Sym A
Model
Result
Result
Result
Result
X
X
X
X
Result

SINTEF

Task III

General feedback on submitted obligs

- A good state machine will in non-trivial cases have several control states
- A state machine should be reactive. This means that each transition requires an input signal to happen. The only exception is the initializing transition from the black ball to the first control state.

General feedback on submitted obligs

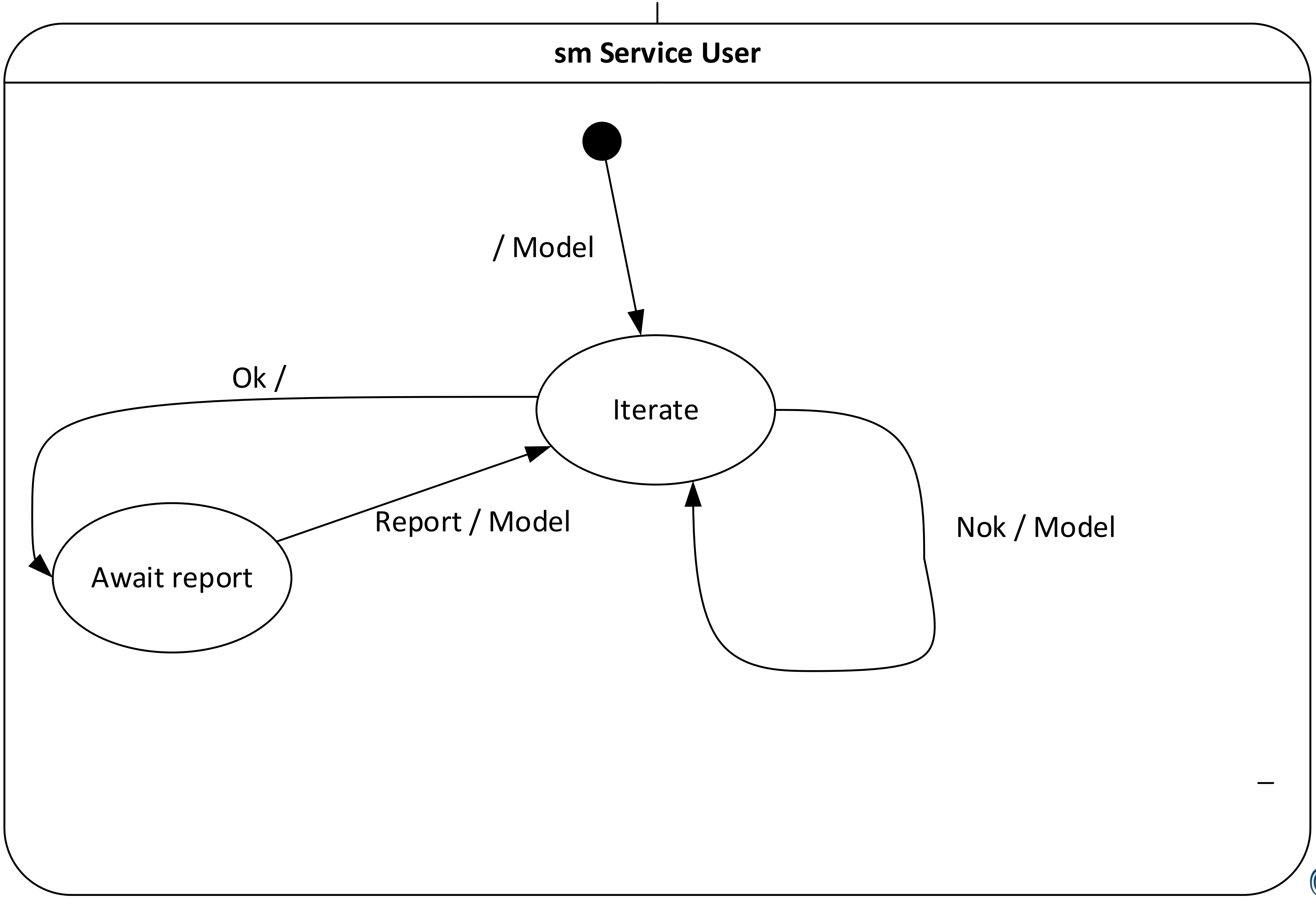
- We use / to distinguish the input signal from the action. The action may do something internal (record a value) and/or send an output signal.
- If a transition has no slash (no /) then the decoration on the transition is an input signal.

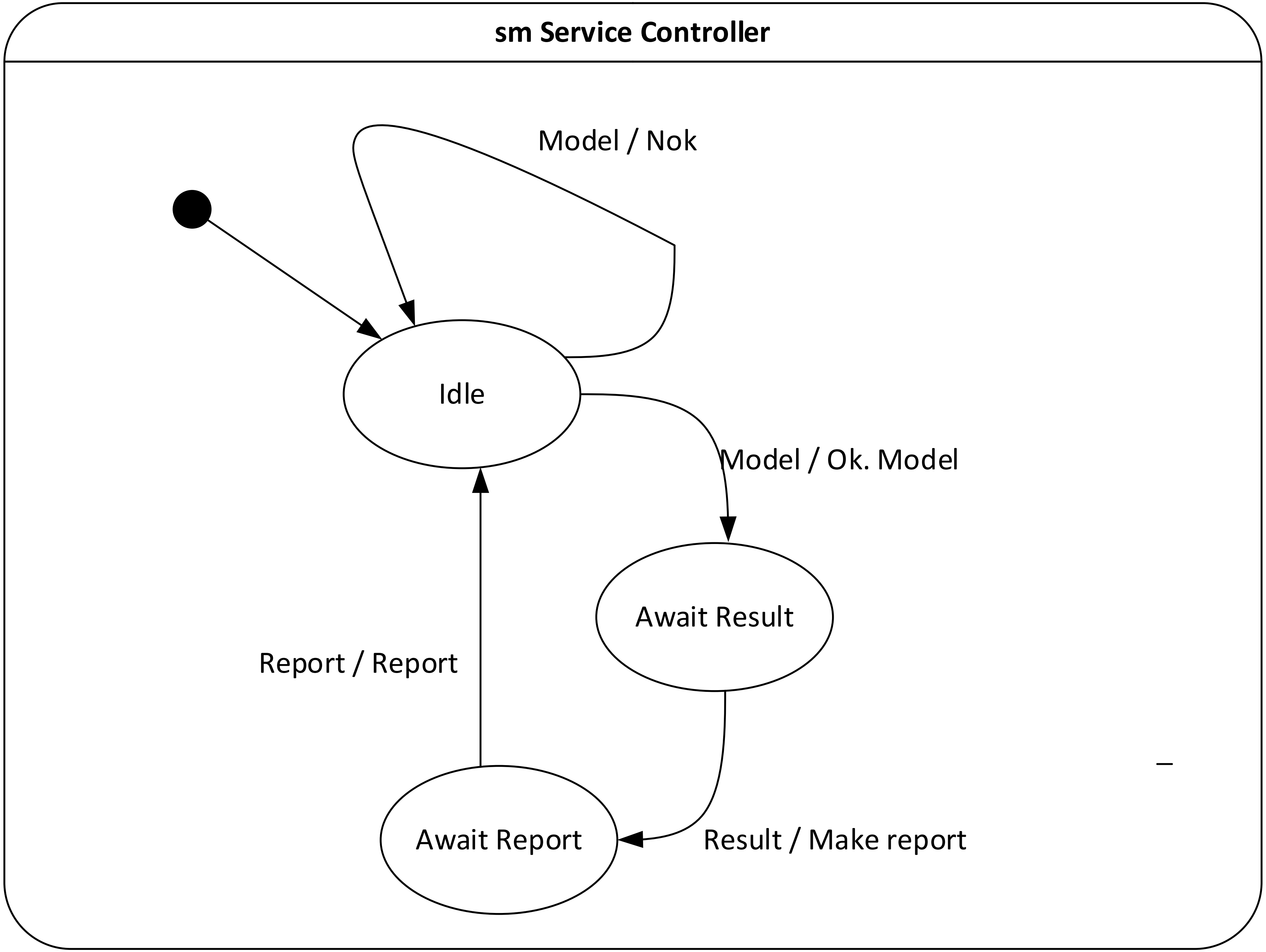
General feedback on submitted obligs

- For the state-machine to be correct wrt its lifeline then it must allow the behaviour (the traces) of the lifeline starting from the black ball.
- This means that the input and output signals of the state machine should be message names from the sequence diagram
- The state-machine may (and will normally) allow many more behaviours.

General feedback on submitted obligs

- The sequence diagram only consider some simple scenarios. Hence, that it is consistent just means that the state machine is ok wrt those.
- Remember that input signals received in a control state for which no transition for this input signal is defined, will be consumed. This means it results in a transition with no output action to itself (remains in the same control state)
- Negative behavior of a state machine is all the traces that the machine cannot produce. Hence, there should be no **veto** constructs (or any other construct for making negative behaviour) in a state machine.


sm Service User
/ Model
Ok /
Iterate
Report / Model
Nok / Model
Await report


sm Service Controller
Model / Nok
Idle
Model / Ok. Model
Await Result
Report / Report
Await Report
Result / Make report

SINTEF

70 år

1950-2020

Teknologi for et bedre samfunn