

Oblig I: Modelling

In this compulsory exercise (Oblig I) you will be trained in modelling the structure and behaviour of processes and technology focusing particularly on interaction and communication. We have selected a case that should be easy to grasp without prior domain knowledge. In a later compulsory exercise (Oblig III) you will be asked to conduct a security risk assessment of the same case.

The description is underspecified, and there are many ways to do the tasks. This is intentional and reflects how things are in practise. You are free to make assumptions as long as they are stated explicitly by you in your solution-paper.

The solution-paper in the form of a single pdf-document should be sent to **kst@sintef.no** by midnight (24:00) September 28. You may work in groups of **maximum three students** or **you may work alone**. Hence, one solution may have up to three names. There should be no collaboration or copying **between different groups**. **Hence, each group should solve the exercise independently.**

You may use the tool of your preference. Some alternatives:

- <https://www.eclipse.org/papyrus/> (powerful but involves a lot to install and use)
- <https://www.draw.io> app (light weight)
- VISIO with UML stencil

Remember to put your real names (not usernames) in the solution-paper so that I can see who wrote it when it is printed on paper.

Case description

Assume some tool-vendor wants to use machine learning to provide improved on-the-fly assistance within the tool to the users of the tool. To preserve confidentiality as well as privacy for the tool-users they intend to use federated machine learning. Federated machine learning allows an algorithm to be trained across multiple decentralized edge devices or servers holding local data samples, without exchanging them. This approach stands in contrast to traditional centralized machine learning techniques where all the local datasets are uploaded to one server, as well as to more classical decentralized approaches which often assume that local data samples are identically distributed. See the following link for further background information:

https://en.wikipedia.org/wiki/Federated_learning

As explained in the above link, there are several different approaches to federated learning. In this oblig we will use centralized federated learning. This means there is a central server at the tool vendor that via the cloud is used to orchestrate the different steps of the algorithms and coordinate all the participating decentralized edge devices or local servers of their tool-customers during the learning process.

We will for simplicity assume the following simplified process:

Initialization: The central server activates the local servers. The local servers wait for the central server to give the calculation tasks.

Client selection: A fraction of local servers is selected by the central server to start training on local data. The selected local servers receive the current statistical model while the others wait for the next federated round.

Configuration: The central server orders selected local servers to undergo training of the model on their local data in a pre-specified fashion (the details of which is not relevant for the oblig).

Reporting: Each selected local servers sends its local statistical model to the central server for aggregation. The central server aggregates the received statistical models and sends back the current statistical model updates to the local servers. The next federated round is started returning to the client selection phase.

Termination: Once a pre-defined termination criterion is met (e.g., a maximum number of iterations or the accuracy of the statistical is greater than a threshold) the central server finalizes the global statistical model, and the process terminates.

Task I

Make a class-diagram describing the relevant concepts. The class-diagram should at least represent the following concepts: Central Server, Vendor, Local Server, Tool, Tool User, Statistical Model, Cloud. The class-diagram should contain the generalization relation and illustrate correct use of both composition and aggregation. All associations should be decorated with multiplicities.

Task II

Make a sequence diagram (or several) describing the interaction in accordance with the above process between Central Server and one Local Server. There should be lifelines for Central Server, Local Server, Central Statistical Model and Local Statistical Model. The sequence diagrams 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.

Task III

- a) Make a state-machine that correctly reflects all the behaviour of the lifeline for the Central Server.
- b) Make a state-machine that correctly reflects all the behaviour of the lifeline for the Local Server.

The two state-machines must allow any behaviour allowed by the respective lifelines in the sequence diagrams but may have additional behaviour.

Remember that the signal names in the state-machines should correspond to message names in the sequence diagrams.