POLITECNICO
MILANO 1863

Software Engineering 2

M Camilli, E Di Nitto, M Rossi

System Description: Copilot

Copilot is a driver-assistance system that helps car drivers with some autonomous operations: lane keeping, cruise control, and emergency braking. To achieve autonomous capabilities, the car is equipped with a combination of sensors and actuators interacting with the software pieces of Copilot to perceive the environment and make driving decisions. In particular, the vehicle has one or more camera sensors(for visual perception) and one or more Lidar sensors (to measure distances). All sensors collect data and periodically wake up Copilot, which can then retrieve the data from them. Copilot logs the data and calculates proper commands to be issued to (external) actuators uniquely identified through an alphanumeric ID. A command includes the ID of the target actuator, a set of numeric values (quantities to be actuated) and a timestamp. The car may be equipped with several actuators; for instance, the Braking System (BS) is expected to reduce the cruise speed up to a certain given value.

System Description: Copilot

Copilot requires the driver to always monitor the driving and be prepared to take control at a moment's notice. The driver can engage (activate) or disengage (deactivate) the autonomous operations at will using hard buttons on the steering wheel. When the autonomous mode is disengaged, Copilot keeps logging data, but it does not issue any commands. When the autonomous mode is engaged, Copilot periodically prompts the driver to take control of the steering wheel by initiating small movements. If the driver responds with a slight movement, the autonomous mode remains engaged. If the driver applies too much force (exceeding a pre-defined threshold), Copilot disengages immediately the autonomous mode. If the driver does not take control of the steering wheel within a certain time frame, Copilot emits an alarm until the driver resumes full control of the car

RASD Question 1

Considering the following goal:

G1: “The driver wants to travel by car with minimal driving effort.”

1) **Define all relevant phenomena** involved in the achievement of the goal. Group the phenomena by category:

- world-only
- shared world-controlled
- shared machine-controlled.

For simplicity, consider sensors and actuators as broad categories, without distinguishing between the specificities of individual sensors and actuators.

RASD Question 2

2) Define **a minimal set of requirements** and domain assumptions that are necessary to achieve the goal.

RASD Question 3

3) Use a UML Class Diagram to describe **the domain of Copilot.**

Design Question 1

Define the **Copilot programming interface** to realize the world-controlled shared phenomena you defined in previous questions.

Interface definition shall include:

- Name of the operation and brief description
- Arguments and corresponding types
- Return type
- Mapping to the corresponding requirement

Design Question 2

Use a UML Component Diagram to **design the structure of Copilot**. The diagram must include all interfaces necessary to realize the operations previously defined and must consider the following information. All sensors collect data and periodically wake up one or more Copilot components called consumer that can retrieve the data from them. The set of consumers include a simple logger (in charge of storing historical data) and a feature coordinator component. The feature coordinator uses one or more driving features which calculate proper commands to be issued to actuators. The currently foreseen features are:

- Automated Lane Keeping (ALK)
- Adaptive Cruise Control (ACC)
- Automated Emergency Braking (AEB).

Based on the commands produced by the driving features, the feature coordinator determines the sequence of commands to be sent to one or more actuators.

Design Question 3

Use **design-level UML Sequence Diagrams** to describe the following two scenarios:

• “The autonomous mode is engaged, and, after some time, it prompts the driver to take control of the steering wheel. The driver responds with a slight movement.”

• “The Lidar sensor signals an obstacle in front of the vehicle; Copilot computes a proper command and then interacts with the Braking System.”

The diagrams must be consistent with the operations and the structure previously defined.

RASD Question 1

World-only phenomena:

- Sensors collect data
- Actuators apply commands
- Driver monitors driving
- Driver continues his/her activities in the real world ignoring the alarm

World-controlled shared phenomena:

- Driver activates autonomous operations
- Driver de-activates autonomous operations
- Driver applies slight force to steering wheel
- Driver applies strong force to steering wheel
- Sensor wakes up system
- Sensor sends data to system (upon request by system)

Machine-controlled shared phenomena:

- System requests data from sensors
- System sends command to actuator
- System prompts driver to take control of vehicle by enacting small movements (e.g., vibrations) on
- steering wheel
- System emits alarm if user does not respond to prompt by applying force to steering wheel

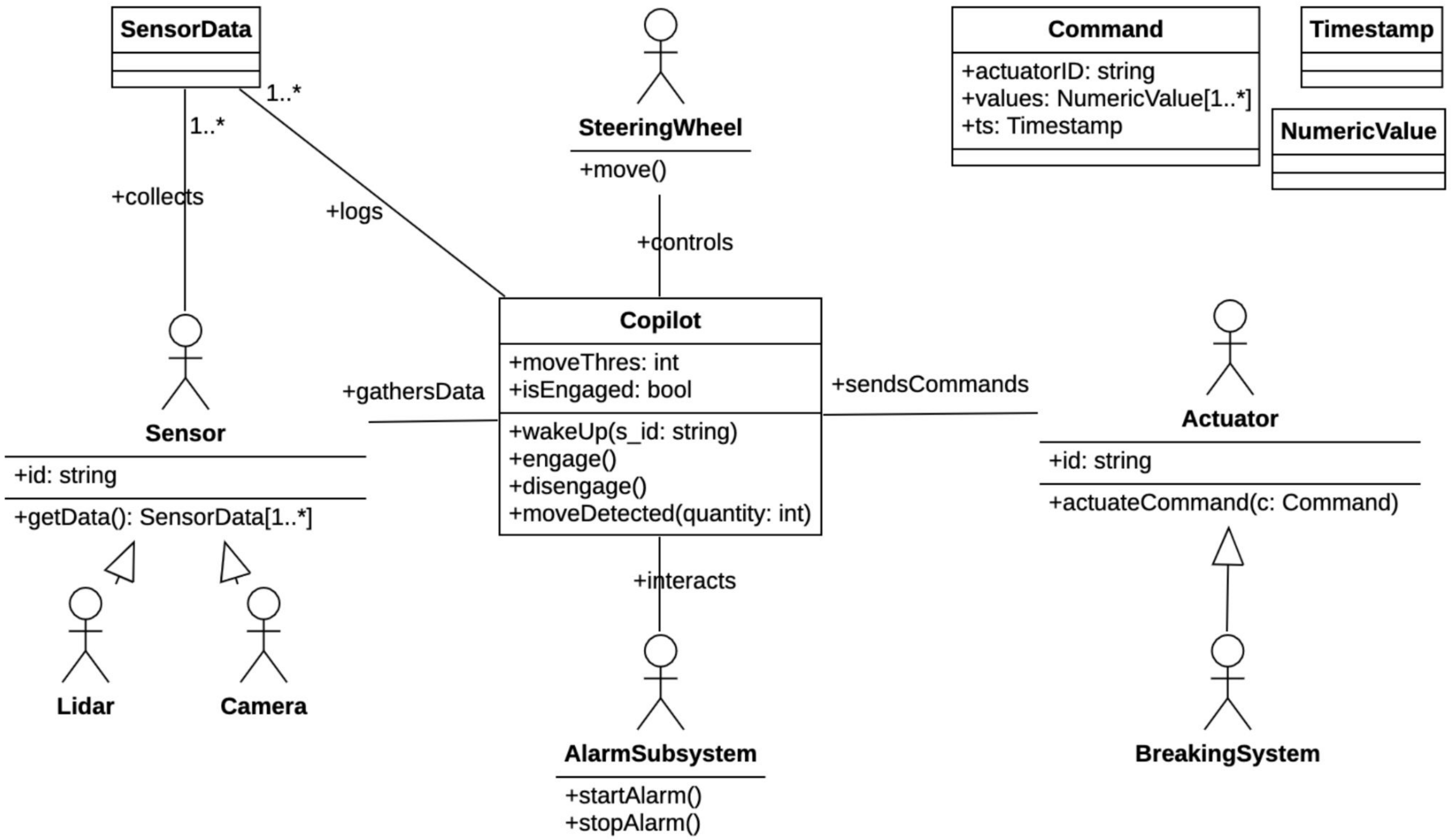
RASD Question 2: Requirements

- **R1**: Copilot shall allow the driver to engage the self-driving mode by using the hard button installed on the steering wheel
- **R2**: Copilot shall allow the driver to disengage the self-driving mode by using the hard button installed on the steering wheel
- **R3**: when a sensor wakes up Copilot, it shall retrieve the data from that sensor if the self-driving mode is engaged
- **R4**: when data is retrieved from sensor(s), Copilot shall compute commands (to be sent to actuators)
- **R5**: when a command is computed, Copilot shall send the set of quantities to the target actuator (e.g., Braking system)
- **R6**: When Copilot is engaged, every T seconds (with T a configuration parameter) Copilot shall prompt the user to apply some force on the steering wheel to check that the driver is attentive; the prompt occurs by making the steering wheel move slightly (for example vibrate)
- **R7**: If, after Copilot prompts the driver, the driver does not apply any force to the steering wheel, Copilot starts emitting an alarm.
- **R8**: After Copilot starts emitting an alarm because the driver did not take control of the steering wheel, it shall stop emitting the alarm only when the driver takes back control of the vehicle.

RASD Question 2: Domain Assumptions

- **A1**: data gathered by sensors are accurate
- **A2**: sensors periodically wake up Copilot
- **A3**: actuators promptly actuate the quantities sent to them
- **A4**: driver eventually takes control of steering wheel after Copilot actuates small movements on the steering wheel or after Copilot emits the alarm.

RASD Question 3



Design Question 1

The operations are essentially those defined for class Copilot in the class diagram of RASD Question 2.

- **wakeUp(s_id: string)**

  This is used by sensors to wake up Copilot when they have data to communicate. The operation takes as input the id of the sensor that wakes up Copilot (which will be used by Copilot to know from which sensors data must be gathered). This corresponds to requirement R3 (notice that the part about Copilot retrieving data from sensors is a machine-controlled phenomenon, so it is not part of this interface).

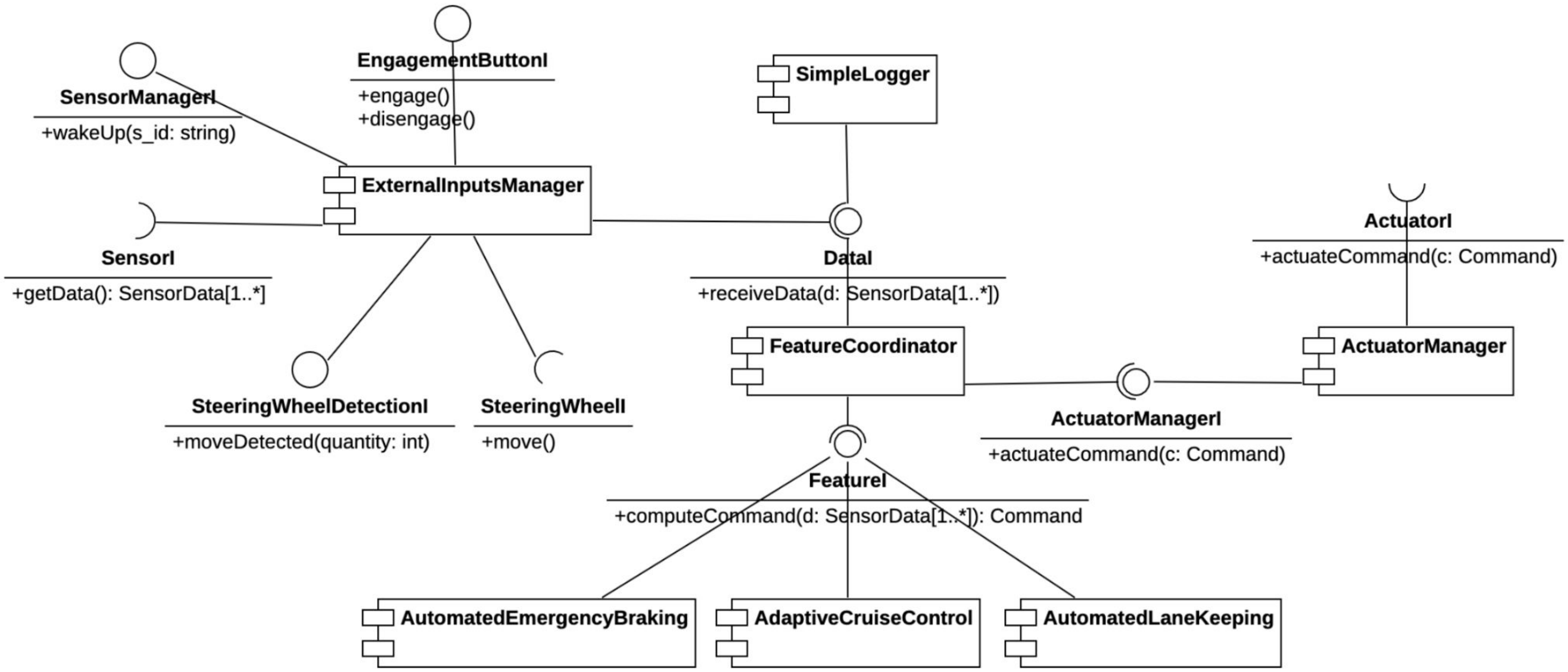
- **engage()/disengage()**

  These are the operations corresponding to the actions performed by the driver to engage and disengage Copilot. They do not take as input any argument. They correspond to requirements R1 and R2.

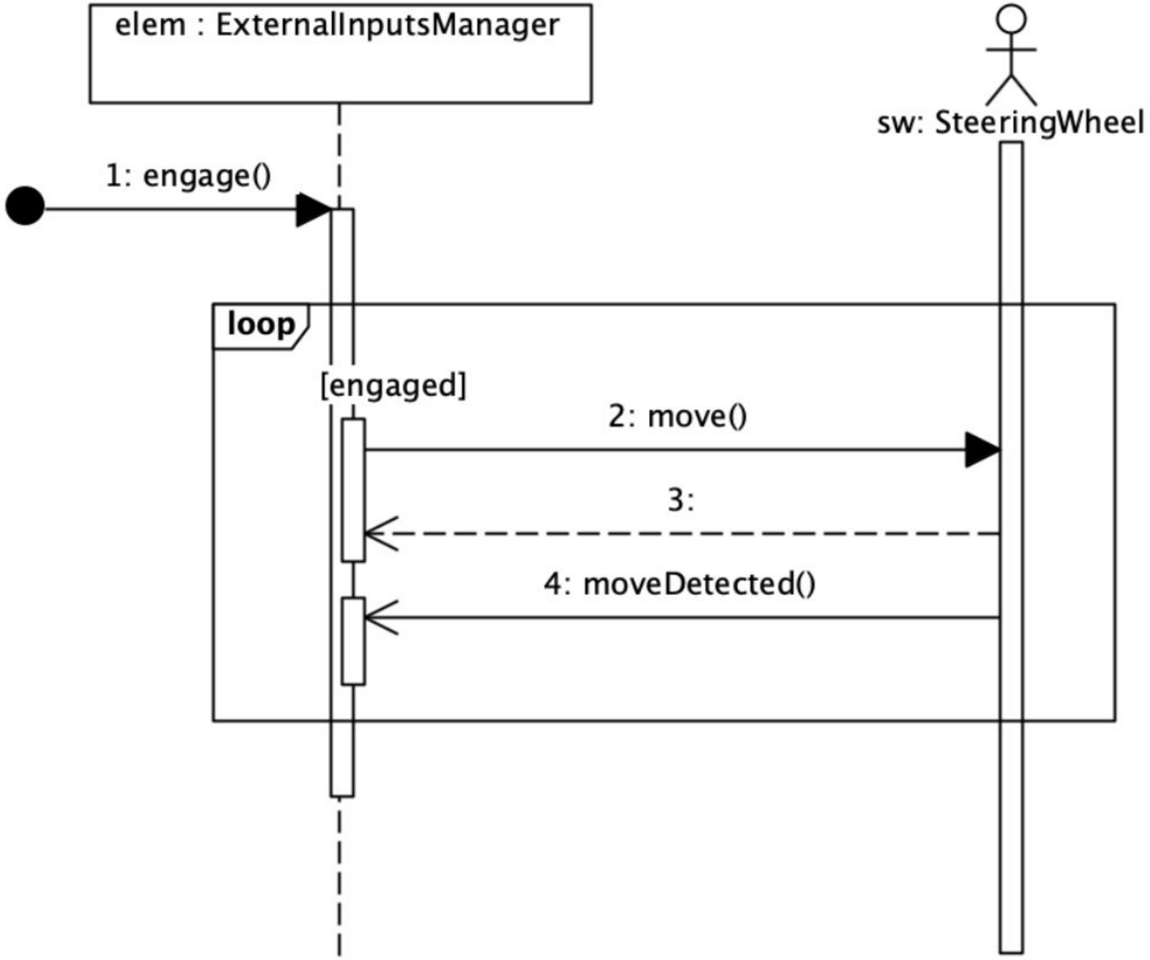
- **moveDetected(quantity : int)**

  This operation is used to communicate to Copilot when the driver applies movements to the steering wheel. It takes as parameter a quantity that corresponds to the force measured on the steering wheel. It is used to implement part of R7 and R8 (the parts that are not about emitting alarms, which are machinecontrolled shared phenomena).

Design Question 2: Component Diagram



Design Question 3: Sequence Diagrams



Design Question 3: Sequence Diagrams