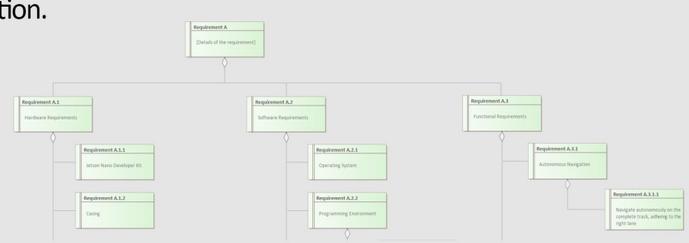
# System Specification Using CONSENS

Hazhir Amiri – Luis Fernando Rodriguez Gutierrez – Leander Hackmann – Furkan Iskender
Fachhochschule Dortmund – Fachbereich Informatik

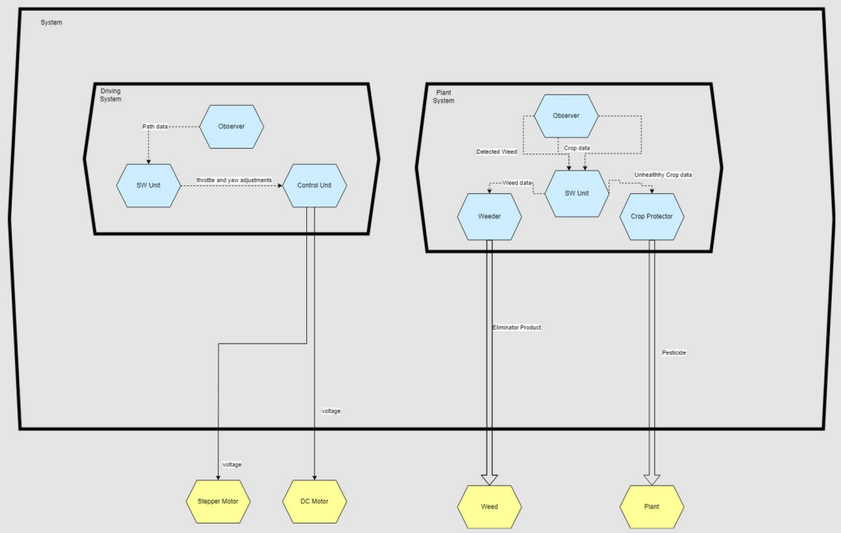
## Requirements

The autonomous robot development with Jetson Nano involves several key categories: hardware, software, functional, testing, and documentation requirements. Hardware needs include a Jetson Nano module, power supply, 32GB MicroSD card, USB Wi-Fi, ventilated casing, two high-resolution cameras, a 1:10 scale vehicle chassis, high-torque motors, motor driver, battery pack, wheels, wiring, breadboard, and mechanisms for weed elimination and fertilizer distribution. Software requirements encompass a Linux-based OS, Python or C++ development, OpenCV, TensorFlow or PyTorch, ROS, and Jetson Inference. Functional requirements cover autonomous navigation, lane detection, plant identification and treatment, and sign recognition. Testing involves performance metrics in both simulations and real-world scenarios. Documentation demands comprehensive user and technical manuals, and a version-controlled repository for source code with thorough comments and documentation.

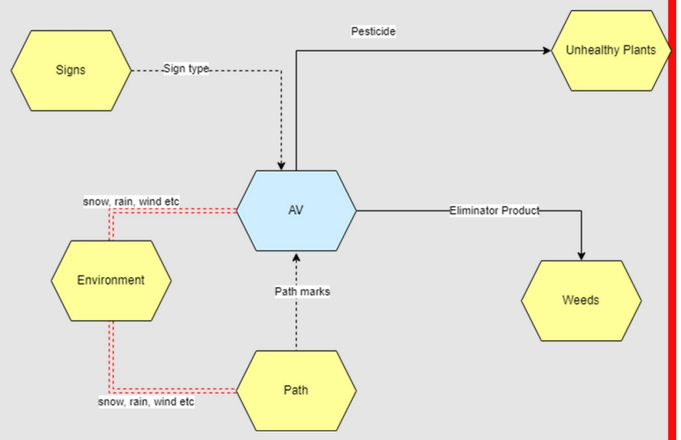



## Active Structure

The active structure includes a model of the system that is designed based on the roles the system and the essential components that the different parts of the system are made of. The system is made up of two main subsystems: the driving system and the plant system. Each system contributes to different, yet essential parts of the whole system, interacting with the outside world.



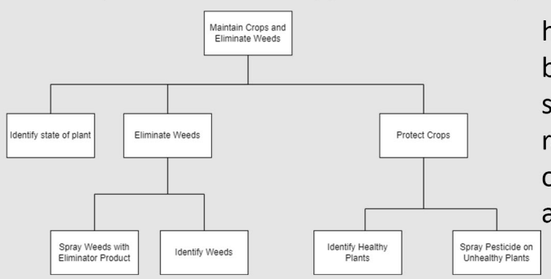

## Environment



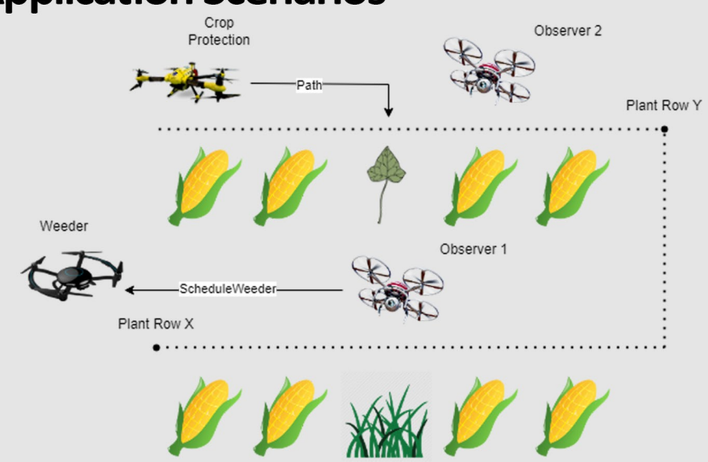

The environment model depicts the interaction of the autonomous vehicle (AV) with its surroundings in the form of effects it receives or causes on the environment. This can be in the form of information it receives from the or the material or energy it sends. Notably, the AV receives vital data from the path in form of path marks. This information can be negatively effected by the environment.

## Functions

The function diagram decomposes the system as top-to-bottom tasks which the fulfillment of each task is accomplished by the success of its sub-tasks. For example, for the function “Protect Crops” to be completed, the two tasks “Identify Healthy Plants” and “Spray Pesticide on Unhealthy Plants” need to be accomplished first. This approach to decomposing the system functions helps the analysis of the basic functions of the systems that need to be realized for the correct operation of the system as a total.




## Application Scenarios




## Shape