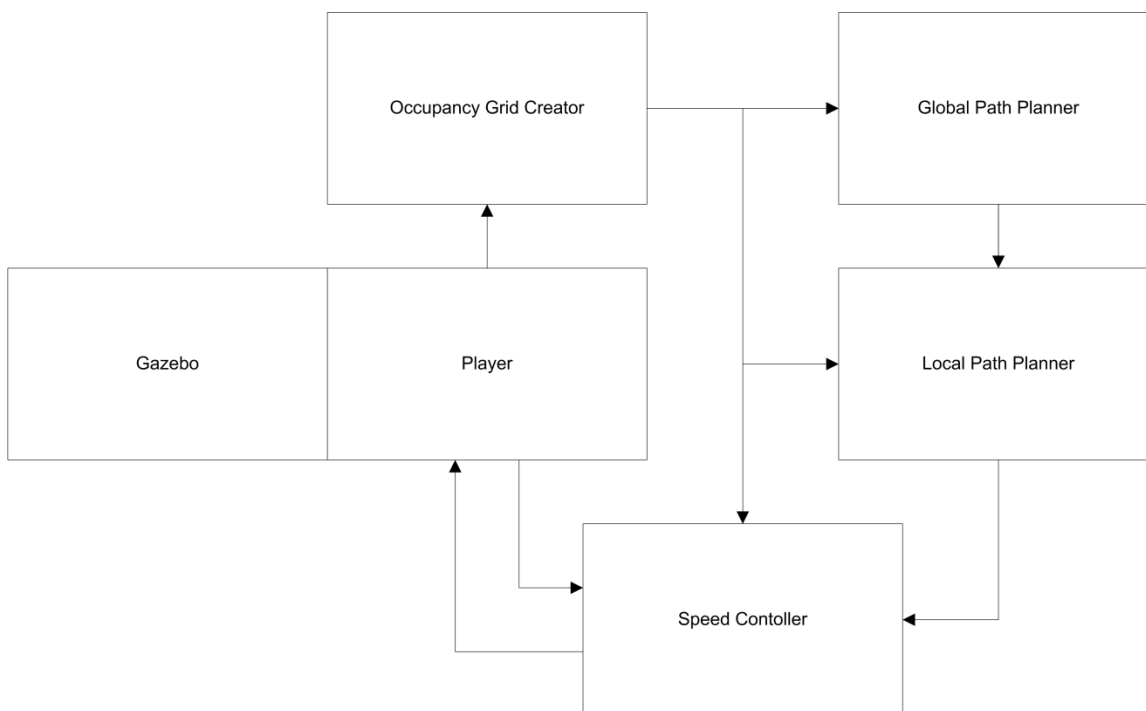


# Report #2

*Rev 1.2*

## High-Level Design



### Occupancy Grid Creator:

- Inputs:** Present location via the Position3D interface from the Player Server  
Laser range information from the Player Server
- Outputs:** Occupancy grid to the Global/Local Path Planner and Speed Controller
- Description:** Creates and maintains an occupancy grid based on past and present data from the laser range finder.

### Global Path Planner:

- Inputs:** Occupancy grid from Occupancy Grid Creator
- Outputs:** List of way-points to Local Path Planner
- Description:** Using the algorithm described in “Flying Fast and Low Among Obstacles: Methodology and Experiments,” S. Scherer, S. Singh, L. Chamberlain, M. Elgersma, *The International Journal of Robotics Research*, vol. 27, pp. 549-574, May 2008. and based on the 2D code created by Dr. Jonathan Eskritt. Uses the Occupancy grid as a map and plots a path to the objective that is a maximum distance from all obstacles.

### Local Path Planner:

- Inputs:** Occupancy grid from the Occupancy Grid Creator  
List of way-points from the Global Path Planner
- Outputs:** Directional information to Speed Controller
- Description:** Either based on the algorithm described in “Flying Fast and Low Among Obstacles: Methodology and Experiments,” S. Scherer, S. Singh, L.

Chamberlain, M. Elgersma, *The International Journal of Robotics Research*, vol. 27, pp. 549-574, May 2008. or using another suitable local planning algorithm such as "nd" or "fvh". Using the list of way-points given by the global path planner, the local path planner smoothes out the path and eliminated redundant way-points. It also reacts to new obstacles discovered on the occupancy grid and corrects the path to avoid them. It passes the nearest way-point's direction to the Speed Controller.

**Speed Controller:**

Inputs: Occupancy grid from the Occupancy Grid Creator  
Directional information from the Local Path Planner  
Current velocity from the Player Server via the Position3D interface.

Outputs: Direction of acceleration via the Position3D interface to the Player Server

Description: The speed controller tells the UAV which direction and how much to accelerate. The desired direction of travel is given from the Local Path Planner and it uses the occupancy grid to determine the maximum safe speed by maintaining a minimum stopping distance to all obstacles. The difference between the current and required velocity is used to determine the appropriate acceleration.

**Gazebo:**

Inputs: World description files  
Robot description files  
Position3D interface information from Player

Outputs: 3D rendering of environment  
Laser range information to Player  
Robot location information via Position3D interface to Player

Description: Gazebo is a robot simulation environment. It provides physics interactions, 3D view of the world, and data to the sensors of the robot.

**Player:**

Inputs: Acceleration commands via the Position3D interface from the Speed Controller  
Laser range information from Gazebo  
Location information via the Position3D interface from Gazebo  
Robot configuration files

Outputs: Laser range information to the Occupancy Grid Creator  
Position3D interface information to Gazebo  
Position3D interface information to Speed Controller

Description: Player is a robot operating system designed to control a real or simulated robot. It provides an interface between our code and the simulated robot devices/sensors.

# Project Schedule

