

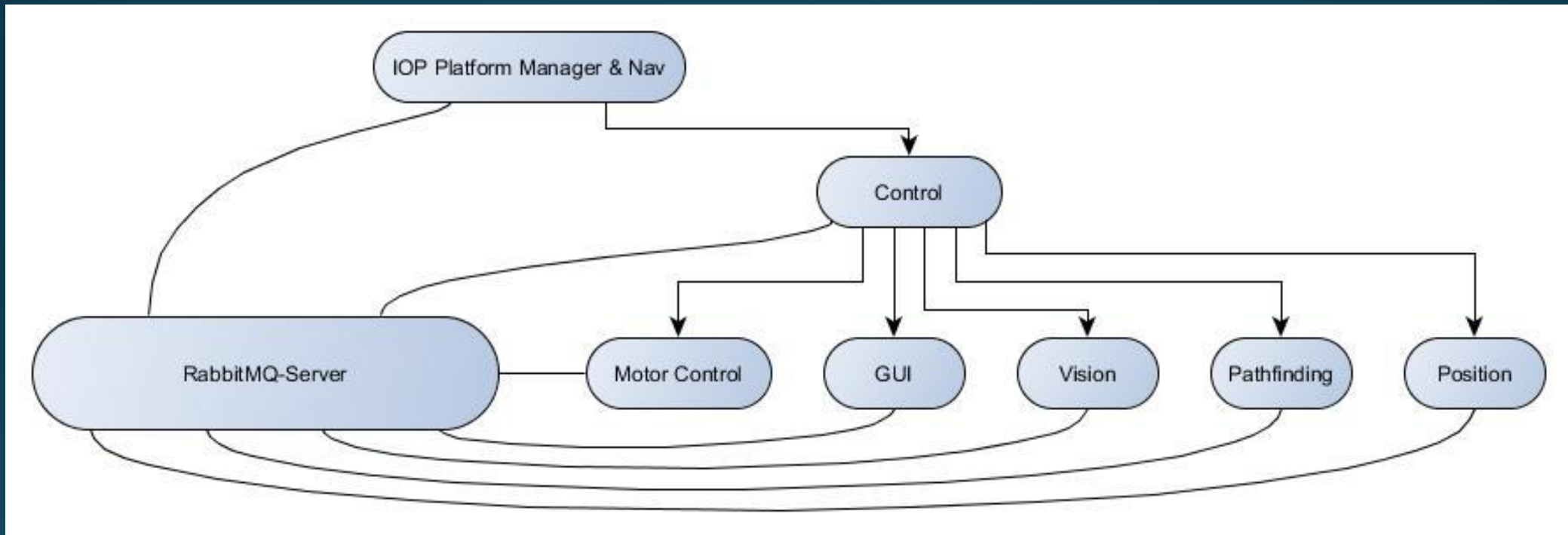
Milestone One

Florida Tech IGVC

Requirements Document

- Intelligent Ground Vehicle Competition (IGVC) rules are the customer we are working for
- Competition rules require the vehicle operate independently without human interaction other than to set up the vehicle for the course
- Software system for the robot will be comprised of independent yet interconnected modules
- Software reliability and stability is a key requirement

Design Document



Test Document

- Reliability is a pivotal part of the operation of the vehicle
- Isolation Testing – Each module will be tested and expected to perform by itself
- Message testing – Each module must be able to communicate using the required messaging service (RabbitMQ) to successfully communicate with the server and other modules
- Software Integration testing will be performed when each module is finished
 - Every software module will be tested with the other modules it interacts with
 - Every software module available will then be tested at the same time
- Hardware integration
 - Full testing of the entire system deployed before vehicle operation if possible
 - Vehicle operation is the final stage of complete integration testing

Pathfinding

- Supporting classes written to simplify actual algorithm, and encapsulate the information pathfinding passes on to other modules
 - Supporting classes need Testing:
 - Vehicle Model
 - Movement Command
 - Geometric Calculations
- First iteration of SBMPO algorithm has been written
 - Untested, currently calls functions that are still stub code
 - Not optimized
 - Uses A* instead of LPA*

GUI

- JavaFX research done
 - Fully understand the API
- Internal synchronization framework developed
 - Must listen to messages, but also keep the "GUI thread" free
- Creation of basic window with all the sections as described in requirements doc
- Minimal work done
 - The same person who is doing the GUI is also doing pathfinding, and ironing out pathfinding algorithm is much higher priority

Vision

- Completed integration of the ZED with the Tegra TX1 after communicating with the vendor to receive firmware updates
- Initial sets of data collected from the ZED for analysis
- Researched potential algorithms for locating obstacles and doing basic line detection
- Set up environment for CUDA, OpenCV and PCL libraries
- Implementation of object detection and line following is in progress

Communication & Simulation

- Finished simulation
- Determined best C++ client library for RabbitMQ
(AMQP-CPP by Copernica Marketing)
- Began writing basic clients in C++
- Debugging Java client

Interoperability (JAUS)

- Standards for communication, testing, and simulation involving remote devices and autonomous vehicles
- Standards set by SAE called JAUS (Joint Architecture for Unmanned Systems)
- UDP connections with remote devices required

Milestone 2 Task Matrix

Task	Will	Adam	Chris	Brent
Finished GUI	20 %	30%	20%	30%
Prototype Navigation	10%	40%	10%	40%
RabbitMQ C++ clients fully implemented	70%	10%	10%	10%
Sensor Position Estimation	20%	30%	30%	20%
Line following and basic obstacle detection	20%	-	80%	-
Prototype Control and IOP	30%	30%	10%	30%