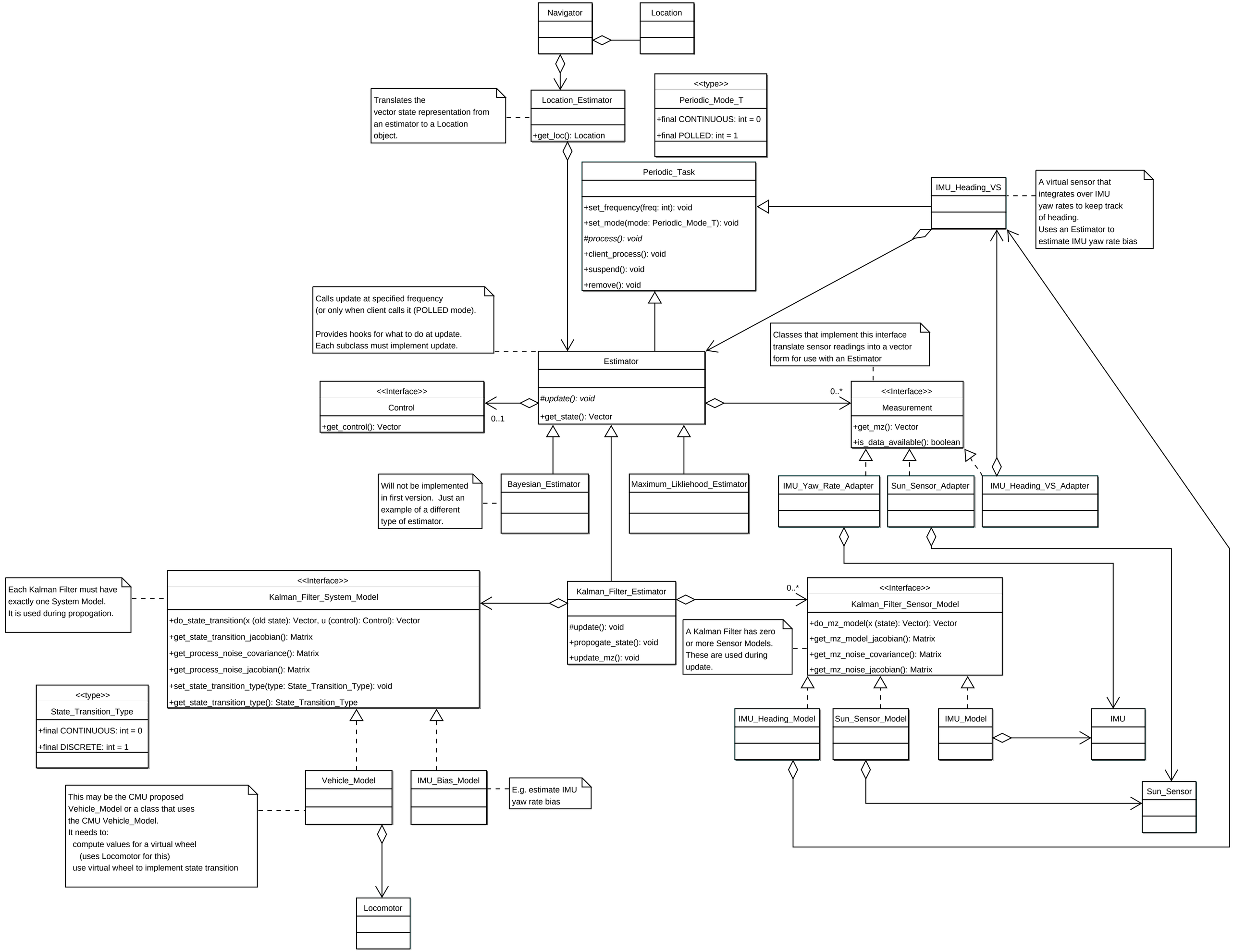
Navigator
Location
Translates the vector state representation from an estimator to a Location object.
Location_Estimator
+get_loc(): Location
<<type>>
Periodic_Mode_T
+final CONTINUOUS: int = 0
+final POLLED: int = 1
Periodic_Task
+set_frequency(freq: int): void
+set_mode(mode: Periodic_Mode_T): void
#process(): void
+client_process(): void
+suspend(): void
+remove(): void
IMU_Heading_VS
A virtual sensor that integrates over IMU yaw rates to keep track of heading. Uses an Estimator to estimate IMU yaw rate bias
Calls update at specified frequency (or only when client calls it (POLLED mode).
Provides hooks for what to do at update. Each subclass must implement update.
Estimator
#update(): void
+get_state(): Vector
Classes that implement this interface translate sensor readings into a vector form for use with an Estimator
<<Interface>>
Control
+get_control(): Vector
0..1
0..*
<<Interface>>
Measurement
+get_mz(): Vector
+is_data_available(): boolean
Will not be implemented in first version. Just an example of a different type of estimator.
Bayesian_Estimator
Maximum_Likliehood_Estimator
IMU_Yaw_Rate_Adapter
Sun_Sensor_Adapter
IMU_Heading_VS_Adapter
Each Kalman Filter must have exactly one System Model. It is used during propogation.
<<Interface>>
Kalman_Filter_System_Model
+do_state_transition(x (old state): Vector, u (control): Control): Vector
+get_state_transition_jacobian(): Matrix
+get_process_noise_covariance(): Matrix
+get_process_noise_jacobian(): Matrix
+set_state_transition_type(type: State_Transition_Type): void
+get_state_transition_type(): State_Transition_Type
Kalman_Filter_Estimator
#update(): void
+propogate_state(): void
+update_mz(): void
0..*
<<Interface>>
Kalman_Filter_Sensor_Model
+do_mz_model(x (state): Vector): Vector
+get_mz_model_jacobian(): Matrix
+get_mz_noise_covariance(): Matrix
+get_mz_noise_jacobian(): Matrix
A Kalman Filter has zero or more Sensor Models. These are used during update.
<<type>>
State_Transition_Type
+final CONTINUOUS: int = 0
+final DISCRETE: int = 1
IMU_Heading_Model
Sun_Sensor_Model
IMU_Model
IMU
Sun_Sensor
Vehicle_Model
IMU_Bias_Model
E.g. estimate IMU yaw rate bias
This may be the CMU proposed Vehicle_Model or a class that uses the CMU Vehicle_Model. It needs to:
compute values for a virtual wheel
(uses Locomotor for this)
use virtual wheel to implement state transition
Locomotor