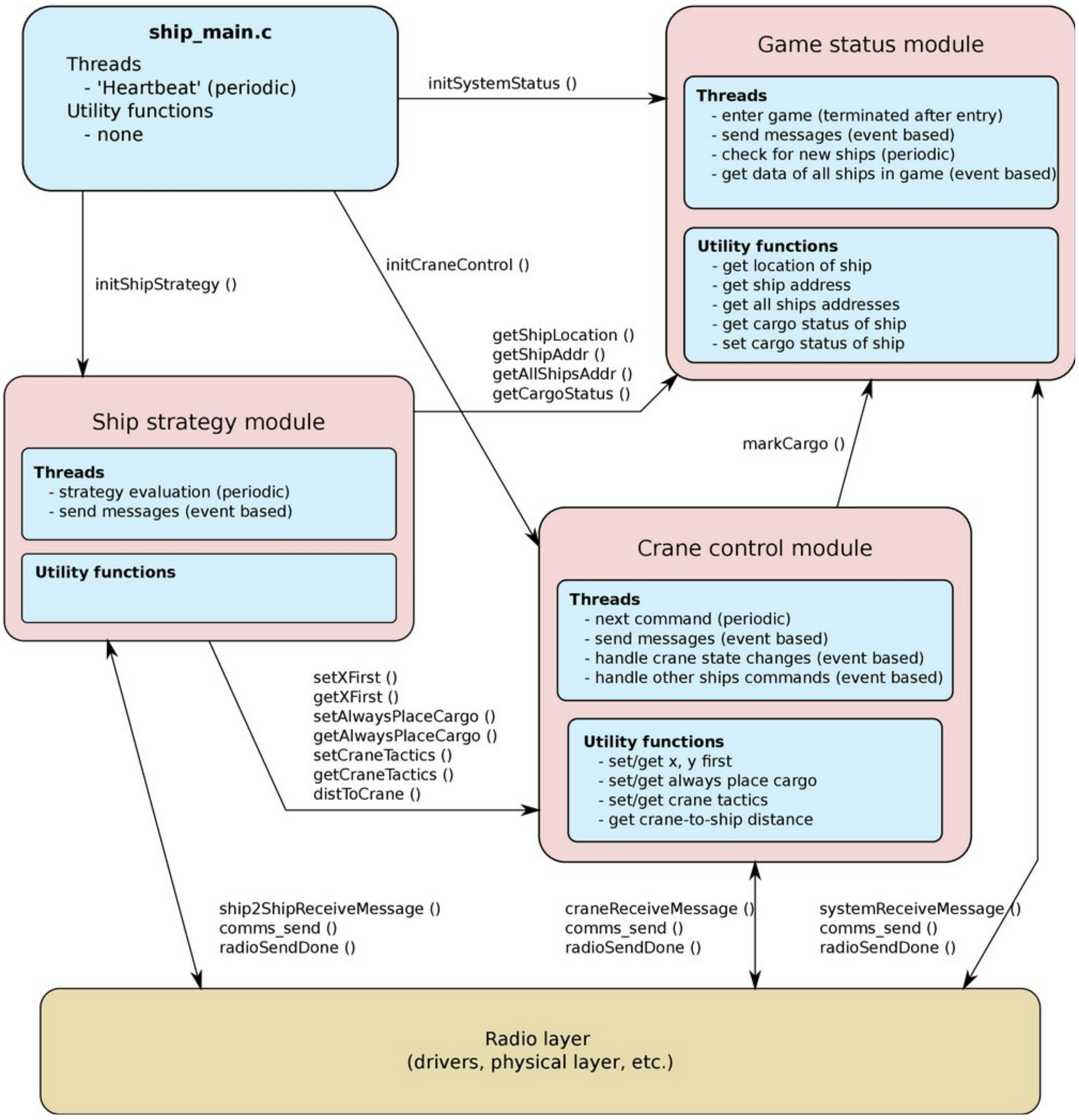
Ship-agent API

Author: Johannes Ehala, ProLab TTÜ, 2021

Documentation of the available functions of the three software modules of the ship-agent.

Table of Contents



Schema 1. Ship-agent software modules and their relations.

Crane control module

This module is responsible for handling message exchange between ship-agent and crane-agent. Command messages are sent to crane-agent once every crane-agent update interval and crane-agent location messages are received and crane-agent state is updated. Crane-agent update interval is defined globally in game_types.h. The command message is sent no sooner than 0.5 seconds before next crane-agent update.

This module also listens to command messages from other ship-agents and stores the latest sent command for each ship-agent. Commands are stored until next crane-agent update, receivement of crane-agent location message clears all stored commands of other ship-agents.

Crane command selection is handled by this module. Command selection is a matter of strategy and tactics. Strategy is implemented and maintained by the ship-agent strategy module, tactics is implemented by this module. The choice of tactics is selected by the strategy module using crane control module API.

A few basic tactical choices are implemented by the crane control module, users may add more implementations. The basic tactical choices are

- setting which direction (x or y) is handled first, when commanding the crane-agent to move to some specified destination
- setting whether to place cargo on any ship whenever the crane-agent is at a ship location or to only place cargo to the tactical objective (ie only to a specified ship and no one else)
- setting current command selection tactic

The default tactic is to get the crane-agent to ship-agents own address. Tactics currently implemented are

- **do nothing** - don't send any crane control command messages
- **to address** - call crane to specified address and place cargo
- **to location** - call crane to specified location and place cargo
- **parrot ship** - send same command message as specified ship
- **popular command** - send the command that is most popular in current crane update round

Crane control module API

The address of a ship is an unique identifier of any ship that has entered the cargo loading game. It is used to distinguish ships and it is also the communication address of ships when sending messages.

void setXFirst (bool val)

@param	val	*true* if crane-agent is to move along x coordinate first; *false* if crane-agent is to move along y coordinate first.

Sets whether crane-agent is commanded to move along x coordinate first or along y coordinate first. This setting is relevant with tactics *cc_to_address* and *cc_to_location.* It has no effect with tactics *cc_do_nothing*, *cc_parrot_ship* and *cc_popular_command*.

bool getXFirst ()

@ return		*true* if crane-agent is to move along x coordinate first; *false* if crane-agent is to move along y coordinate first.

Returns whether crane-agent is commanded to move along x coordinate first or along y coordinate first. This setting is relevant with tactics *cc_to_address* and *cc_to_location*. It has no effect with tactics *cc_do_nothing*, *cc_parrot_ship* and *cc_popular_command*.

void setAlwaysPlaceCargo (bool val)

@param	val	*true* results in always issuing place cargo command when crane-agent is at the location of a ship; *false* results in placing cargo only at location designated by current tactic.

Sets whether cargo is always placed on any ship whenever the crane-agent happens to be at the location of a ship. This setting is relevant with tactics *cc_to_address* and *cc_to_location*. It has no effect with tactics *cc_do_nothing*, *cc_parrot_ship* and *cc_popular_command*.

NB! Cargo placement command is issued only if the ship does not have cargo yet.

bool getAlwaysPlaceCargo ()

@return		*true* indicates always issuing place cargo command when crane-agent is at the location of a ship; *false* indicates placing cargo only at location designated by current tactic.

Returns cargo placement tactics choice. This setting is relevant with tactics *cc_to_address* and *cc_to_location*. It has no effect with tactics *cc_do_nothing*, *cc_parrot_ship* and *cc_popular_command*.

NB! Cargo placement command is issued only if the ship does not have cargo yet.

void setCraneTactics (cmd_sel_tactic_t tt, am_addr_t ship_addr, loc_bundle_t loc)

@param	tt	command selection tactic; predefined tactic choices are *cc_to_address*, *cc_to_location*, *cc_do_nothing*, *cc_parrot_ship* and *cc_popular_command*
@param	ship_addr	address of a ship; relevant with tactics *cc_to_address and cc_parrot_ship*
@param	loc	location of a ship; relevant with tactics *cc_to_location*

Sets tactical choice for crane-agent command selection.

NB! Find *loc_bundle_t* definition at game_types.h file.

cmd_sel_tactic_t getCraneTactics (am_addr_t *ship_addr, loc_bundle_t *loc)

@param	*ship_addr	address of the ship; relevant with tactics *cc_to_address and cc_parrot_ship*
@param	*loc	location of the ship; relevant with tactics *cc_to_location*

@return		command selection tactic; predefined tactic choices are *cc_to_address*, *cc_to_location*, *cc_do_nothing*, *cc_parrot_ship* and *cc_popular_command*

Returns current tactical choice for crane-agent command selection. Depending on tactic the parameters *ship_addr* and *loc* are set to relevant destination ship information.

NB! Find *loc_bundle_t* definition at game_types.h file.

uint16_t distToCrane (loc_bundle_t loc)

@param	loc	a location from where to measure distance to crane-agent
@return		distance to crane-agent

Returns distance from location *loc* to crane-agent current location. A distance of zero means that the crane-agent is at location *loc*.

NB! Find *loc_bundle_t* definition at game_types.h file.

Game status module

The game status module of the ship-agent is responsible for keeping track of game status. Game status includes game time and information about ships (including self) and their status (communication address, location, cargo status, departure time). The module is a database of information about ships active in the current game. The database can hold *MAX_SHIPS* number of ships (see *game_types.h*).

Game status module sends different query messages to crane-agent and receives response messages from crane-agent in order to keep its database up to date. This is done automatically in the background. In good radio transmission conditions (ie no packet loss), the database would be kept up to date to within one second. As a back-up a full game state query is done every *GS_UPDATE_INTERVAL* seconds to compensate for any lost messages earlier. The update interval is set in game_status.c file and its default value is 60 seconds.

Game status module API

The address of a ship is an unique identifier of any ship that has entered the cargo loading game. It is used to distinguish ships and it is also the communication address of ships when sending messages.

loc_bundle_t getShipLocation (am_addr_t ship_addr)

@param ship_addr	the address of the ship whose location is returned
@return	location of the ship

Returns location of ship with address *ship_addr.* If no such ship is found returns 0 for both coordinates.

NB! Find *loc_bundle_t* definition at game_types.h file.

void markCargo (am_addr_t addr)

@param	addr	the address of the ship whose cargo has been received

Marks cargo status as true for ship with address *addr*, if such a ship is found in game database. Use with care! There is no revers command to mark cargo status false. Cargo status is tracked by crane control and game state module and generally the user should not need to call this function.

uint8_t getAllShipsAddr (am_addr_t saddr[], uint8_t mlen)

@param	saddr[]	buffer to store addresses of ships currently in the game
@param	mlen	size of buffer *saddr*
@return		number of ships currently in game

Fills buffer pointed to by *saddr* with addresses of all ships currently known (ships in the game). Returns the number of ships added to buffer *saddr*.

am_addr_t getShipAddr (loc_bundle_t sloc)

@param	sloc	a location on the game plane
@return		address of the ship in location *sloc* or zero if no ship in that location

Returns address of the ship in location *sloc* or zero if no ship in that location.

NB! Find *loc_bundle_t* definition at game_types.h file.

cargo_status_t getCargoStatus (am_addr_t ship_addr)

@param	ship_addr	the address of the ship whose cargo status to check
@return		the status of cargo; possible values *cs_cargo_received*, *cs_cargo_not_received*, *cs_unknown_ship_addr*

Returns cargo status of ship with address *ship_addr*. Possible return values:

- *cs_cargo_received* - cargo has been received, cargo present
- *cs_cargo_not_received* - cargo has not been received, cargo not present
- *cs_unknown_ship_addr* - ship not in database, unknown ship

NB! Find *cargo_status_t* definition at game_status.h file.

Ship strategy module

This module establishes the ship-agents' strategy for getting its cargo loaded. The strategy is created by each user, there are no example strategies to choose from. In most cases the strategy should involve setting up some sort of cooperation between the ship-agents currently in the game. How ever the choice of strategy is up to the user and cooperation is not a requirement.

Ship strategy module implements its chosen strategy by creating a message exchange between ship-agents (if necessary) and by selecting suitable tactics for the crane command module. See the crane command module API for available tactics. New tactics may be added to the crane command module if necessary.

Message exchange between ship-agents is handled by this module. Two types of messages have been implemented, the user should implement additional messages needed to support the chosen strategy.

Ship strategy module API

The address of a ship is an unique identifier of any ship that has entered the cargo loading game. It is used to distinguish ships and it is also the communication address of ships when sending messages.

void sendNextCommandMsg (crane_command_t cmd, am_addr_t dest)

@param	cmd	the crane command to send to ship-agent with address *dest*. Possible values are *cm_no_command, cm_up, cm_down, cm_left, cm_right, cm_place_cargo, cm_current_location, cm_nothing_to_do*, see crane_command_t type in game_types.h file.
@param	dest	address of the ship-agent to send this message to. Use value *AM_BROADCAST_ADDR* to send to every ship-agent in the game.

Sends a message with a crane command value to other ship-agents. The meaning behind this message (whether it is a suggestion for the next command, or the next command of the sender, or something else) is defined by the user.

void sendNextShipMsg (am_addr_t ship_addr, am_addr_t dest)

@param	ship_addr	address of a ship-agent to send to ship-agent with address *dest*.
@param	dest	address of the ship-agent to send this message to. Use value *AM_BROADCAST_ADDR* to send to every ship-agent in the game.

Sends a message with a ship-agent address value to other ship-agents. The meaning behind this message (whether it is a suggestion for the next ship to be loaded, or the ship to select as leader, or something else) is defined by the user.