MAES Application programming interface

“A Multi-Agent-based Framework for Embedded System (MAES) Library”

Student:
Carmen Chan Zheng

Master of Science Embedded Systems
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Delft, The Netherlands

July 6, 2017

1 API Enum types

- AGENT_MODE values:
  - ACTIVE
  - SUSPENDED
  - WAITING
  - TERMINATED
  - NO_MODE
- ERROR_CODE values:
  - NO_ERROR
  - FOUND
  - HANDLE_NULL
  - LIST_FULL
  - DUPLICATED
  - NOT_FOUND
  - TIMEOUT
  - INVALID
  - NOT_REGISTERED
- MSG_TYPE
  - ACCEPT_PROPOSAL
  - AGREE
  - CANCEL
  - CFP
  - CONFIRM
  - DISCONFIRM
  - FAILURE
  - INFORM
  - INFORM_IF
  - INFORM_REF
  - NOT_UNDERSTOOD
  - PROPAGATE
  - PROPOSE
  - QUERY_IF
  - QUERY_REF
  - REFUSE

  - REJECT_PROPOSAL
  - REQUEST
  - REQUEST_WHEN
  - REQUEST_WHENEVER
  - SUBSCRIBE
  - NO_RESPONSE
- REQUEST_ACTION
  - REGISTER
  - DEREGISTER
  - KILL
  - RESUME
  - SUSPEND
  - MODIFY
  - BROADCAST
  - RESTART
- ORG_AFFILIATION
  - OWNER
  - ADMIN
  - MEMBER
  - NON_MEMBER
- ORG_ROLE
  - MODERATOR
  - PARTICIPANT
  - VISITOR
  - NONE
- ORG_TYPE
  - HIERARCHY
  - TEAM

2 Agent class

The module `Agent` provides the API for Agent object building.

2.1 Constructor `Agent(String name, int pri, char *AgentStack, int sizeStack)`

Default constructor.

Parameters
`String name` - Agent's name.
`int priority` - Agent's priority.
`char *AgentStack` - Agent's stack
`int sizeStack` - Agent's stack size

3 Agent_Msg class

The module `Agent_Msg` provides API message object creation and manipulation.

3.1 Constructor `Agent_Msg()`

Default constructor.

3.2 Methods details

- `ERROR_CODE add_receiver(Agent_AID aid_receiver)`
  Adds receiver to the multicast list.
  Parameters
  `Agent_AID aid_receiver` - Agent's AID to be added to the list.
  Returns
  Error code defined in section 1.

- `ERROR_CODE remove_receiver(Agent_AID aid_receiver)`
  Removes receiver from the multicast list.
  Parameters
  `Agent_AID aid_receiver` - Agent's AID to be removed from the list.
  Returns
  Error code defined in section 1.

- `void clear_all_receiver()`
  Clears multicast list.
  Parameters
  None
  Returns
  `NULL`

- `void refresh_list()`
  Updates list with registered agents. Removes agent if not registered or if does not belong same organization as the caller agent.

Parameters
None
Returns
`NULL`

- `MSG_TYPE receive(Uint32 timeout))`
  Receives message from other agents. Suspends execution of the agent until a message is received or the timeout is expired.
  Parameters
  `Uint32 timeout` - Maximum waiting time specified in system's ticks.
  `BIOS_WAIT_FOREVER` causes agent to wait indefinitely. `BIOS_NO_WAIT` returns immediately.
  Returns
  Message type defined in section 1

- `ERROR_CODE send(Agent_AID aid_receiver, int timeout)`
  Sends message to specific receiver. Only can be sent if sender and receiver belong to the same organization or do not belong to any organization.
  Parameters
  `Agent_AID aid_receiver` - Target agent's AID.
  `int timeout` - Maximum waiting time specified in system's ticks.
  `BIOS_WAIT_FOREVER` causes agent to wait indefinitely for a slot in the receiver mailbox. `BIOS_NO_WAIT` returns immediately.
  Returns
  Error code defined in section 1.

- `ERROR_CODE send()`
  Sends message to all receivers in the multicast list.
  Parameters
  None.
  Returns
  Returns last error type defined in section 1.

- `void set_msg_type(MSG_TYPE type)`
  Sets message type.
  Parameters
  `MSG_TYPE type` - set to any value defined in section 1
  Returns
  `NULL`

- `void set_msg_string(String body)`
  Sets the string content of the message.
  Parameters
  `String body` - String content of the message
  Returns
  `NULL`

- `void set_msg_int(int content)`
  Sets the integer content of the message.
  Parameters
  `int content` - Integer content of the message
  Returns
  `NULL`

- `MsgObj *get_msg()`

Gets the MsgObj.
Parameters
None
Returns
Pointer to the `MsgObj` object.

- `MSG_TYPE get_msg_type()`
  Gets the message type.
  Parameters
  None
  Returns
  Message type defined in section 1.

- `int get_msg_string()`
  Gets the string content of the message.
  Parameters
  None
  Returns
  String content of the message.

- `int get_msg_int()`
  Gets the integer content of the message.
  Parameters
  None
  Returns
  Integer content of the message.

- `Agent_AID get_sender()`
  Gets sender's AID of the message.
  Parameters
  None
  Returns
  Sender's AID.

- `Agent_AID get_target_agent()`
  Gets target agent AID of the message.
  Parameters
  None
  Returns
  Target agent's AID.

- `ERROR_CODE request_AP(REQUEST_ACTION request, Agent_AID target_agent)`
  Request the Agent Platform to perform any of these services: Register, Deregister, Resume, Suspend .
  Parameters
  `REQUEST_ACTION request` - Request type specified in section 1.
  `Agent_AID target_agent` - Target agent whose service is performed on.
  Returns
  Error code specified in section 1.

- `ERROR_CODE modify_pri(Agent_AID target_agent, int pri)`
  Request the Agent Platform to perform Modify service.
  Parameters
  `Agent_AID target_agent` - Target agent whose service is performed on.
  `int priority` - Priority to be changed in the target agent.

Returns
Error code specified in section 1.

- `ERROR_CODE kill(Agent_AID & target_agent)`
  Request the Agent Platform to perform Kill service.
  Parameters
  `Agent_AID &target_agent` - Target agent to be killed
  Returns
  Error code specified in section 1.

- `ERROR_CODE broadcast(String content)`
  Request the Agent Platform to perform Broadcast service.
  Parameters
  `String content`: Content to be broadcast. **Returns**
  Error code specified in section 1.

- `ERROR_CODE restart()`
  Request the Agent Platform restart the caller agent.
  Parameters
  `NONE`
  Returns
  Error code specified in section 1.

4 Agent_Platform class

The module `Agent_Platform` provides API for Agent Platform construction, initialization and agent management services.

4.1 Constructor `Agent_Platform`

Constructors.

- `Agent_Platform(String name)` - Sets local Agent Platform name.

- `Agent_Platform(String name,USER_DEF_COND*user_cond)` - Sets local Agent Platform name. Set developer's customized conditions.

4.2 Methods details

- `bool boot()`
  Initializes Agent Platform. Creates AMS agent with task stack size of 4KB. Only can be called from main()
  Parameters
  None
  Returns
  `TRUE` if AMS agent is created successfully. System aborts if AP has failed to initialize.

- `void agent_init(Agent &a, Task_FuncPtr behaviour, Agent_AID &aid)`
  Initializes Agent object. Creates the Agent's AID and its mailbox. Only can be called from main()
  Parameters
  `Agent &a` - Agent object to be initialized.

`Task_FuncPtr behaviour` - Wrapper behaviours function to be passed to the agent.
`Agent_AID &aid` - Output paramenter. When the agent is created succesfully, the value of the variable is different than null.
Returns
`TRUE` if AMS agent is created successfully. System aborts if AP has failed to initialize.

- `void agent_init(Agent &a, Task_FuncPtr behaviour, Agent_AID &aid, UArg arg0, UArg arg1)`
Initializes Agent object. Creates the Agent's AID and its mailbox. Only can be called from main()
Parameters
`Agent &a` - Agent object to be initialized.
`Task_FuncPtr behaviour` - Wrapper behaviours function to be passed to the agent.
`Agent_AID &aid` - Output paramenter. When the agent is created succesfully, the value of the variable is different than null.
`UArg arg0, arg1` - Arguments to be passed to the function module.
Returns
`TRUE` if AMS agent is created successfully. System aborts if AP has failed to initialize.

- `bool agent_search(Agent_AID aid)`
Search agent within Agent Platform.
Parameters
`Agent_AID aid` - Agent's AID to be searched.
Returns
`TRUE` if found.

- `void agent_wait(Uint32 ticks)`
Caller agent suspends during time specified by user.
Parameters
`Uint32 ticks` - System ticks time to be suspended.
Returns
`NULL`

- `void agent_yield()`
Caller agent yields the processor to another active agent of equal priority. If there is no other same-priority agent is active, no effect is seen.
Parameters
None. **Returns**
`NULL`

- `Agent_AID get_running_agent()`
Gets the AID of the current running agent
Parameters
None.
Returns
Current running agent's AID

- `AGENT_MODE get_state(Agent_AID)`
Gets state of an agent
Parameters
`Agent_AID` - The agent's AID whose state is returned.
Returns
Returns state defined in section 1.

- `Agent_info get_Agent_description(Agent_AID aid)`
Gets agent description

Parameters
`Agent_AID` - The agent's AID whose description is returned.
Returns
Agent's description structure.

- `AP_Description get_AP_description()`
  Gets Agent Platform description whose caller agent is living.
  Parameters
  None.
  Returns
  Agent Platform description structure.

- `ERROR_CODE register_agent(Agent_AID aid)`
  Register agent in the Agent Platform. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `Agent_AID aid` - Agent's AID to be registered.
  Returns
  Error code described in section 1.

- `ERROR_CODE deregister_agent(Agent_AID aid)`
  Deregister agent in the Agent Platform. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `Agent_AID aid` - Agent's AID to be deregistered.
  Returns
  Error code described in section 1.

- `ERROR_CODE kill_agent(Agent_AID aid)`
  Kill agent in the Agent Platform. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `Agent_AID aid` - Agent's AID to be killed.
  Returns
  Error code described in section 1.

- `ERROR_CODE suspend_agent(Agent_AID aid)`
  Suspend agent in the Agent Platform. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `Agent_AID aid` - Agent's AID to be suspended.
  Returns
  Error code described in section 1.

- `ERROR_CODE resume_agent(Agent_AID aid)`
  Resume agent in the Agent Platform. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `Agent_AID aid` - Agent's AID to be resumend.
  Returns
  Error code described in section 1.

- `ERROR_CODE modify_agent_pri(Agent_AID aid, int pri)`
  Modify agent's priority. Method only restricted for the use of the highest priority agent (AMS agent)

Parameters
`Agent_AID aid` - Agent's AID to be modified.
Returns
Error code described in section 1.

- `void deregister_agent(MsgObj *msg)`
  Broadcast a msg object to all the members of the Agent Platform. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `MsgObj *msg` - Message to be broadcast.
  Returns
  `NULL`
- `void restart(Agent_AID aid)`
  Restart agent's execution and stack. Method only restricted for the use of the highest priority agent (AMS agent)
  Parameters
  `Agent_AID aid` - Agent's AID to be restarted.
  Returns
  Error code described in section 1.

5 USER_DEF_COND class

The module `USER_DEF_COND` provides API for the developer to create customized conditions so AMS services can be performed.

5.1 Methods details

- `virtual bool register_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to register an agent.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.
- `virtual bool kill_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to kill an agent.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.
- `virtual bool deregister_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to deregister an agent.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

- `virtual bool suspend_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to suspend an agent.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

- `virtual bool resume_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to resume an agent.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

- `virtual bool modify_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to modify an agent.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

- `virtual bool broadcast_cond()`
  Method to be overridden by the developer. Customized conditions to be met in order to broadcast a message.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

6 Behaviour classes

The Behaviour module `USER_DEF_COND` provides API for the developer to create customized behaviours for an agent.

- `Generic_Behaviour()`
  Default constructor.

- `void execute()`
  Executes the action(), setup() and done() is a defined order.
  Parameters
  None.
  Returns
  `NULL`

- `virtual void action()`
  Method to be overridden by the developer. Body of the behaviour. Computation code to be executed.
  Parameters
  None.

Returns
`NULL`

- `virtual void setup()`
  Method to be overridden by the developer. Executed once at the beginning.
  Parameters
  None.
  Returns
  `NULL`

- `virtual bool done()`
  Method to be overridden by the developer. Executed after action.
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

- `virtual bool failure_detection()`
  Method to be overridden by the developer. Executed after action().
  Parameters
  None.
  Returns
  `TRUE` if condition is satisfied.

- `virtual void failure_identification()`
  Method to be overridden by the developer. Executed if failure_detection() returns TRUE.
  Parameters
  None.
  Returns
  `NULL`

- `virtual void failure_recovery()`
  Method to be overridden by the developer. Executed after failure_identification.
  Parameters
  None.
  Returns
  `NULL`

6.1 OneShotBehaviour class

- `OneShotBehaviour()`
  Default constructor.

- `void execute()`
  Executes the action(), setup() and done() is a defined order.
  Parameters
  None.
  Returns
  `NULL`

- `virtual void action()`
  Method to be overridden by the developer. Body of the behaviour. Computation code to be executed.
  Parameters
  None.

Returns
`NULL`

- `virtual void setup()`
  Method to be overridden by the developer. Executed once at the beginning.
  Parameters
  None.
  Returns
  `NULL`

6.2 Cyclic Behaviour class

- `CyclicBehaviour()`
  Default constructor.
- `void execute()`
  Executes the action(), setup() and done() is a defined order.
  Parameters
  None.
  Returns
  `NULL`
- `virtual void action()`
  Method to be overridden by the developer. Body of the behaviour. Computation code to be executed.
  Parameters
  None.
  Returns
  `NULL`
- `virtual void setup()`
  Method to be overridden by the developer. Executed once at the beginning.
  Parameters
  None.
  Returns
  `NULL`

7 Agent_Organization class

7.1 Constructor `Agent_organization(ORG_TYPE organization_type)`

Default constructor. Initializes object with organizatio type specified in section 1.

7.2 Methods details

- `ERROR_CODE create()`
  Creates Agent Organization. Caller agent is assigned as the organization owner. If this already have owner, then returns error.
  Parameters
  NONE.

Returns
Error code described in section 1.

- `ERROR_CODE destroy()`
  Destroys Agent Organization. Only can be called by the owner of the organization. Releases all the agents of the organization
  Parameters
  NONE.
  Returns
  Error code described in section 1.

- `ERROR_CODE change_owner(Agent_AID aid)`
  Changes organization's owner. Only can be called by the owner of the organization
  Parameters
  `Agent_AID aid` - new organization's owner
  Returns
  Error code described in section 1.

- `ERROR_CODE isMember(Agent_AID aid)`
  Searches member within organization
  Parameters
  `Agent_AID aid` - agent's aid to be searched.
  Returns
  Error code described in section 1.

- `ERROR_CODE isBanned(Agent_AID aid)`
  Verify if an agent is banned from the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be looked within the banned list.
  Returns
  Error code described in section 1.

- `ERROR_CODE set_admin(Agent_AID aid)`
  Sets organization's administrator. Only can be called by the owner of the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be added as administrator.
  Returns
  Error code described in section 1.

- `ERROR_CODE set_moderator(Agent_AID aid)`
  Sets organization's moderator. Only can be called by the owner of the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be added as moderator.
  Returns
  Error code described in section 1.

- `ERROR_CODE add_agent(Agent_AID aid)`
  Adds new member to the organization. Only can be called by the owner or admin of the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be added.
  Returns
  Error code described in section 1.

- `ERROR_CODE kick_agent(Agent_AID aid)`
  Kicks member from the organization. Only can be called by the owner or admin of the organization.

Parameters
`Agent_AID aid` - agent's aid to be kicked.
Returns
Error code described in section 1.

- `ERROR_CODE ban_agent(Agent_AID aid)`
  Bans member from the organization. Only can be called by the owner or admin of the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be banned.
  Returns
  Error code described in section 1.

- `ERROR_CODE remove_ban(Agent_AID aid)`
  Removes ban from a banned agent. Only can be called by the owner or admin of the organization.
  Parameters
  `Agent_AID aid` - agent's aid ban to be removed.
  Returns
  Error code described in section 1.

- `void clear_ban_list()`
  Removes all the agents from the banned list. Only can be called by the owner or admin of the organization.
  Parameters
  NONE. **Returns**
  `NULL`.

- `ERROR_CODE set_participant(Agent_AID aid)`
  Gives voice to an agent inside of the organization. Only can be called by the owner or moderator of the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be set as participant.
  Returns
  Error code described in section 1.

- `ERROR_CODE set_visitor(Agent_AID aid)`
  Removes voice to an agent inside of the organization, it only can receive messages. Only can be called by the owner or moderator of the organization.
  Parameters
  `Agent_AID aid` - agent's aid to be set as visitor.
  Returns
  Error code described in section 1.

- `ORG_TYPE get_org_type()`
  Gets organization's type.
  Parameters
  NONE.
  Returns
  Organization type described in section 1.

- `org_info get_info()`
  Gets organization's information.
  Parameters
  NONE.
  Returns

Organization info structure.

- `int get_size()`
  Gets number of members within organization.
  Parameters
  NONE.
  Returns
  Organization's number of member.

- `MSG_TYPE invite(Agent_Msg msg, int password, Agent_AID target_agent, int timeout)`
  Invites an agent to join the organization
  Parameters
  `Agent_Msg msg` - message object with the invitation content.
  `int password` - password for entering the organization.
  `Agent_AID target_agent` - agent to be invited.
  `int timeout` - Maximum waiting time for response specified in system's ticks.
  `BIOS_WAIT_FOREVER` causes agent to wait indefinitely. `BIOS_NO_WAIT` returns immediately.
  Returns
  Message type specified in 1