MiRACLE
MiRACLE is an application layer protocol intended for use within MissionEDU to standardize requests, set rules for connections and define message structure.

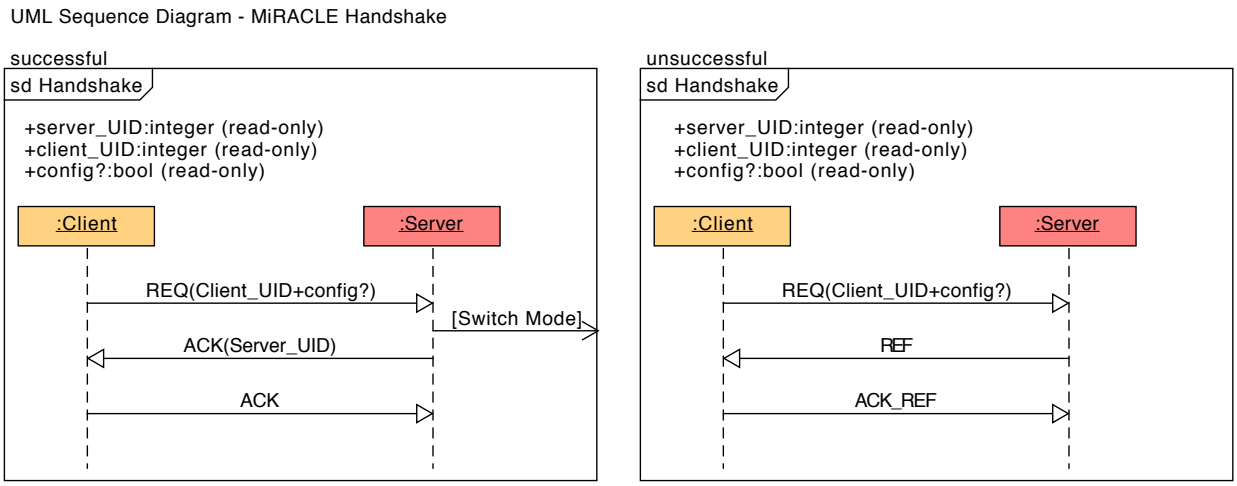
Service
MissionEDU allows to perform remote execution of methods stored on a robotics controller. When establishing a connection with the tablet client it receives a list of available commands and a list of tasks for students created by the teacher. These commands can be used within the tasks to create programs with a graphical user interface. An execution request is sent to the server which confirms it and sends the result of the command back to the client.

A configuration client accesses the server to create and change methods (which have to be compiled), add and change tasks and to reboot the server.

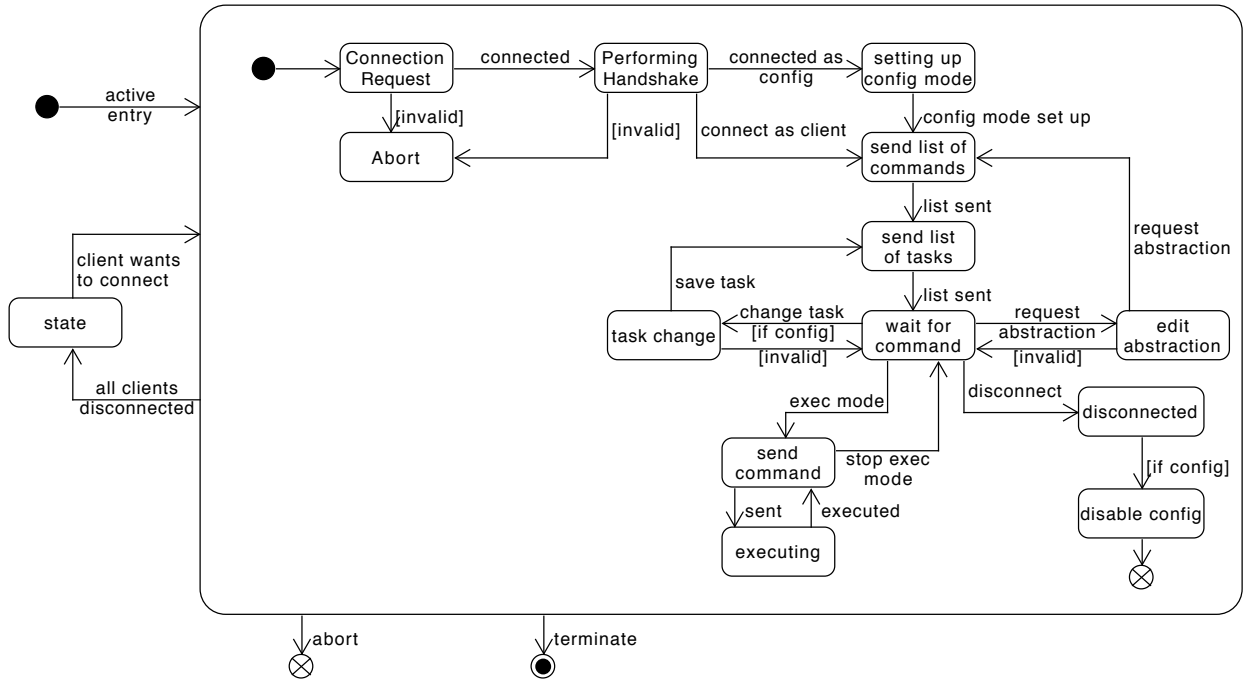
External functionality
The protocol provides messages to perform a three-way-handshake, get the list of functions, send execution requests, send execution confirmations, send execution results, change of methods, creation of methods, deletion of methods, change of tasks, addition of tasks, deletion of tasks, recompilation of methods, restarting of the server, disconnecting.

Internal functionality
Dependant on implementation.

Protocol Specification



Handshake, failed and successful.



State-Transition Diagram of the MiRACLE protocol

Data format definition

Message Types

Handshaking
REQ, ACK, REF, ARF

Disconnecting
DSC
Payload: None

Command List
CML
Payload: [{<CommandName>, <CommandReturn>, "CommandParams":
[{<CommandParamName>, <CommandParamDataType>, <CommandParamRanges>:
{<min>,<max>}}]}]
NOTE: Contains a block "Code" in config mode

Task List
TSL
Payload: [{<TaskUID>, <TaskName>, <TaskDesc>: {<ShortDesc>, <LongDesk>},
<CommandsAvailable>: [<CommandName>]}]

Start Execute Mode
SEM
Payload: None

End Execute Mode
EEM

Payload: {state: ""} //state: successful, error, ended

Send Execute Command
SEC
Payload: {<CommandName>, commandParams: [{<CommandParamName>, <CommandParam>}]}

Execution Command Response
ECR
Payload: {<ReturnType>, <Response>}

Execution Command Error
ECE
Payload: {<ErrorDescription>}

Request Abstraction Layer
RAL
Payload: None

Update Abstraction Layer
UAL
Payload: {<AbstractionLayerFile>}

Abstraction Layer Error
ALE
Payload: {<ErrorDescription>}

Add Command
ACM
Payload: {<CommandName>, <CommandReturn>, "CommandParams": [{<CommandParamName>, <CommandParamDataType>, <CommandParamRanges>: {<min>,<max>}}]}

Change Command
CCM
Payload: {<CommandName>, <CommandReturn>, "CommandParams": [{<CommandParamName>, <CommandParamDataType>, <CommandParamRanges>: {<min>,<max>}}]}

Delete Command
DCM
Payload: {<CommandName>}

Add Task
ATS
Payload: {<TaskName>, taskDesc: {<ShortDesc>, <LongDesk>}, commandsAvailable: [<CommandName>]}

Change Task
CTS
Payload: {<TaskUID>, <TaskName>, taskDesc: {<ShortDesc>, <LongDesk>}, commandsAvailable: [<CommandName>]}

Delete Task
DTS
Payload: {<TaskUID>}

Task Error
TSE
Payload: {<ErrorDescription>}

Command Error
CER
Payload: {<ErrorDescription>}

_____
Message Layout A

```
{
      "MessageType":"",
      "Payload" : {

      }
}
```

Message Layout B

```
{
      "MessageType":"",
      "Payload" : [

      ]
}
```

ToDo:
Generate versions for abstractions, submit these versions, sign versions, determine client type