FUSION FOR ENERGY

MARTe2 Users Meeting Messages

Andre Neto, Filippo Sartori

May, 2019

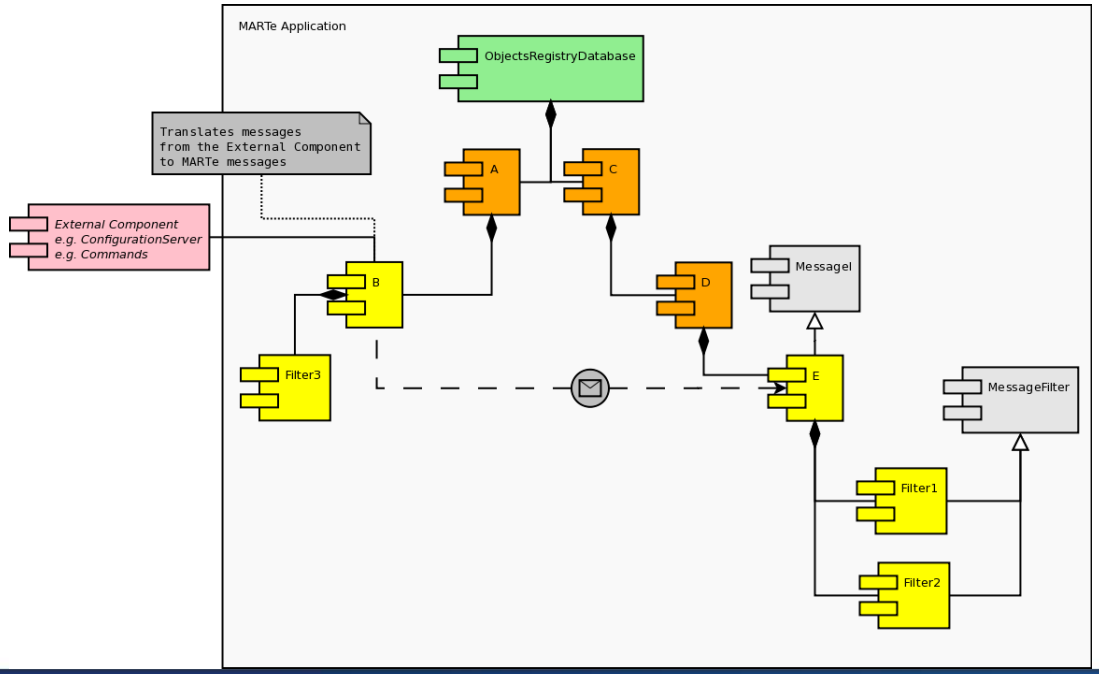
Messages

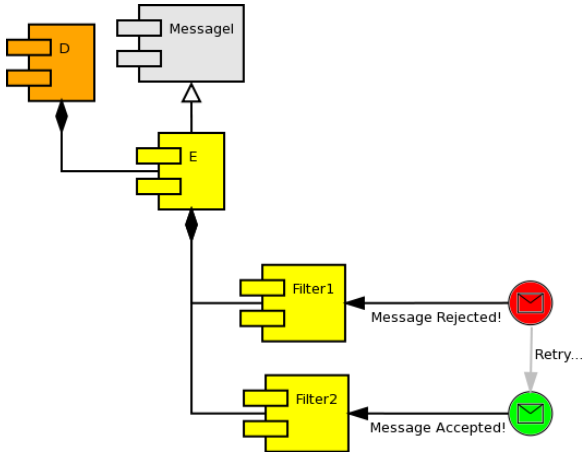
Allows components to exchange information and commands based on the Objects addresses (source and destination) in the ObjectRegistryDatabase

- Change the behaviour of an application based only on configuration data
  - i.e. without requiring any code recompilation.
- Provide a generic interface between MARTe components and any components and protocols that live outside a MARTe application
  - Deployment of applications into new plants without changing code
- Typically used for non real-time activities, such as configuration and state management

MessageI

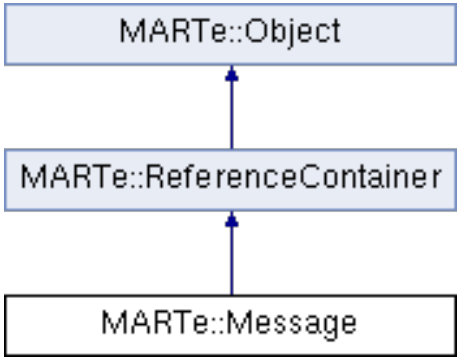
- In order to be able to receive messages a component shall inherit from **MessageI**
- The component shall register one or more filters that will be responsible for deciding if a given message is to be accepted.
- Filters shall inherit from **MessageFilter** and can be either permanent or temporary.

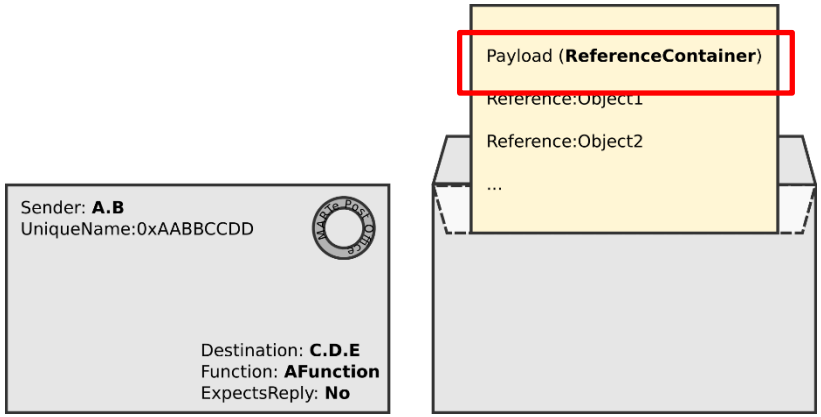




Message

- **Sender**: the Object that is sending the message;
- **Destination**: the Object that will receive the message;
- **Function**: a string which identifies the scope of the action to be performed in the Destination object;
- **IsReply**: true if this message is a reply to a previous message;
- **ExpectsReply**: true if a reply is expected from the Destination component.



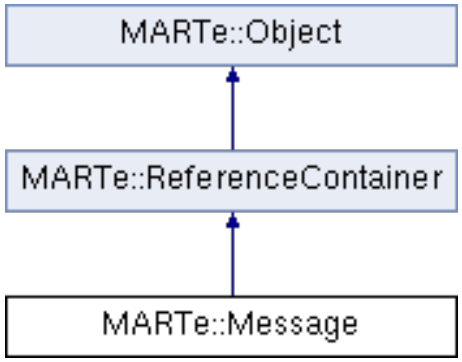


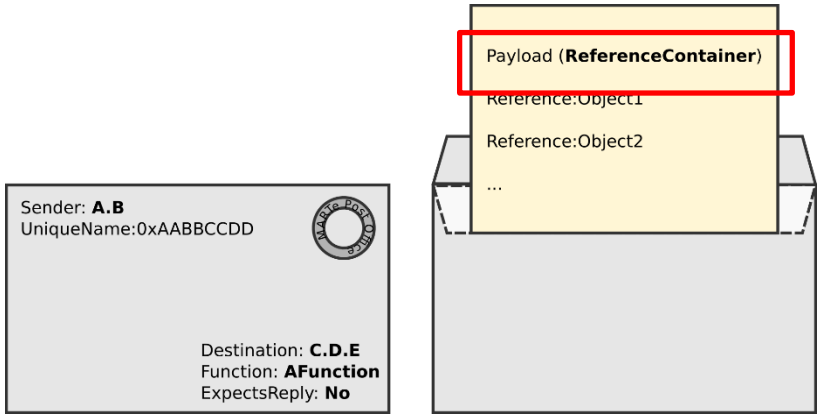
Note

Given that the Message is also a ReferenceContainer the payload is defined by the References that are inserted into the container

Replies

- The reply to a message can be either **direct** or **indirect**
  - **Direct:** sent in threading context of the caller (blocking)
  - **Indirect:** replied in the threading context of the receiver (non-blocking)
    - Requires **caller** to implement **MessageI**

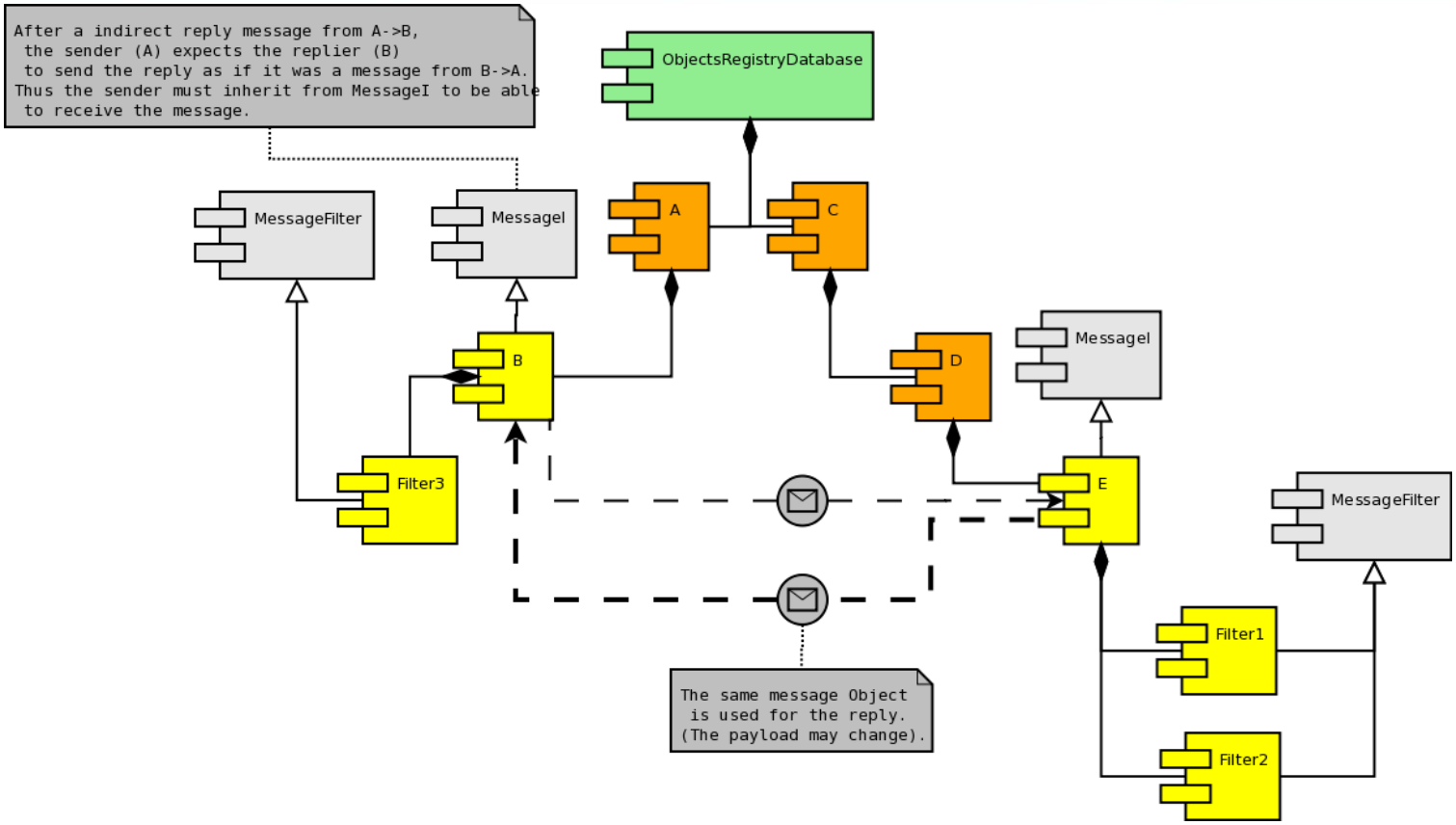




Note

The replier is allowed to change the payload as needed.

Replies



Note

The message sender does not know what filter will be used to consume the message. As such, it cannot make any assumptions on how the message will be handled (e.g. threading context, existence of a queue, …).

Note

The type of filter **does not depend** on the fact of the message being sent with a **direct** or an **indirect** reply.

Remote function calls

Enable functions to be called using messages

- Functions have to be registered

```
ErrorManagement::ErrorType (*) ();
ErrorManagement::ErrorType (*) (T param1);
ErrorManagement::ErrorType (*) (T param1, T param2);
ErrorManagement::ErrorType (*) (T param1, T param2, T param3);
ErrorManagement::ErrorType (*) (T param1, T param2, T param3, T param4);
```

- Parameters are encoded in a Message by adding a ConfigurationDatabase node, named Parameters

```
+Msg1 = {
  Class = Message
  Destination = "MsgRec1"
  Function = "Function1"
  Mode = "ExpectsIndirectReply"
  +Parameters = {
    Class = ConfigurationDatabase
    param1 = 2
    param2 = 3.14
  }
}
```

Exercises

- Function registration & RPC

Function registration

Objective: register functions that can be called using messages

- Modify Other/ControllerEx2/ControllerEx2.cpp
- Modify the ResetGain1 method to set to zero the values of
  - gains1.lowGains.gain1
  - gains1.highGains.gain1
  - gains2.lowGains.gain1
  - gains2.highGains.gain1

```
cd ~/Projects/MARTe2-demos-padova/
export MARTe2_DIR=~/Projects/MARTe2-dev
export MARTe2_Components_DIR=~/Projects/MARTe2-components/
make -f Makefile.x86-linux
```

- Compile
- Run the application

```
cd ~/Projects/MARTe2-demos-padova/Startup/
./Main.sh -l Loader -f ../Configurations/Messages-1.cfg -m
MyController1:ResetGain1
```

Success: application executes and all the *.gain1 are set to zero

Function registration II

Objective: register functions with parameters that can be called using messages

- Modify Other/ControllerEx2/ControllerEx2.cpp
- Create a new function named **UpdateGain1** which accepts **4** float32 as input and update the following values accordingly
  - gains1.lowGains.gain1
  - gains1.highGains.gain1
  - gains2.lowGains.gain1
  - gains2.highGains.gain1
- Compile

```
cd ~/Projects/MARTe2-demos-padova/
export MARTe2_DIR=~/Projects/MARTe2-dev
export MARTe2_Components_DIR=~/Projects/MARTe2-components/
make -f Makefile.x86-linux
```

- Run the application

```
cd ~/Projects/MARTe2-demos-padova/Startup/
./Main.sh -l Loader -f ../Configurations/Messages-2.cfg -m StateMachine:CHANGE
```

Success: application executes and all the *.gain1 are set to 1,2,3 and 4 respectively

FUSION FOR ENERGY

Thank you for your attention

Follow us on:

www.f4e.europa.eu

www.twitter.com/fusionforenergy

www.youtube.com/fusionforenergy

www.linkedin.com/company/fusion-for-energy

www.flickr.com/photos/fusionforenergy