PlantUML

Scripting UML Diagrams
UML Sequence Diagram...

A Sequence Diagram

myActor

My Superfast Database

alt

[Synchronous messages]

1 sync request

2 another sync request

[Synchronous messages]

3 async request

4 response

this is the response to the async request

myActor

My Superfast Database

Different arrow types
... from a Script

```text
@startuml

title A Sequence Diagram

actor myActor

database "My Superfast Database" as myDb

note right of myDb: in the script we use the abbreviation

autonumber

alt Synchronous messages

myActor -> myDb: sync request

else Another synchronous messages

myActor -->[#black] myDb: another sync request

else Asynchronous messages

myActor -->[#red][#black] myDb: async request

myActor //[#green]-- myDb: response

note right: this is the response to the async request

end

caption Different arrow types

@enduml
```

© 2016 Georg Hellack
# Supported Diagram Types

**Structural Diagrams**

- Component
- Class
- Deployment
- Object

**Behavioural Diagrams**

- Use Case
- Sequence
- Activity
- State
Use Case Diagram

@startuml
actor User
actor Administrator as admin
User <|-- admin
(Use Database) as (Use) << Main >>
User --> (Use)
admin --> (Use)
@enduml
@startuml
package "Collections" {
abstract class AbstractList
IList ()- AbstractList
AbstractList <|-- ArrayList
class ArrayList {
    Object[] elementData
    size()}
}

class MyClass
MyClass *- ArrayList
@enduml
@startuml
| App |
start
:call lib function;
| Library |
:validate data;
if(is valid?) then (no)
| App |
:error message;
else (yes)
| Library |
:prepare recordset;
| Database |
:prepare update;
:commit changes;
endif
| App |
stop
@enduml
State Diagram

@startuml

state "Tea Timer Off" as Toff
state "Timer is running" as Trun
state "Timer is ringing" as Tring

[*] --> Toff
Toff --> Trun: 'Start' button pressed
Trun --> Trun: timer not expired
Trun --> Tring: timer expired
Tring --> [*]

@enduml
Applications using PlantUML

Consult http://plantuml.com/running.html for updated list and links to the different tools.
Architecture

Online Editors (e.g. PlantText)

Doc Generators (e.g. Doxygen)

Word Template (e.g. Doxygen)

plantuml.jar

Graphviz
dot.exe
Online Editor

A Sequence Diagram

myActor

My Superfast Database

in the script we use the abbreviation

alt [Synchronous messages]
1 sync request

alt [Another synchronous messages]
2 another sync request

alt [Asynchronous messages]
3 async request
4 response

this is the response to the async request

myActor

My Superfast Database

Different arrow types
Doxygen

- PlantUML script placed within Doxygen comments in source code
- Doxygen executes PlantUML and places images in generated doc
- It is no longer necessary to parse the PlantUML code separately and add references to images in Doxygen comments!
- path to plantuml.jar must be specified in Doxygen configuration
/**
 * @brief A sample class
 */

class ClassA
{
    public:
    /**
     * @brief A sample method
     *
     * @startuml
     * title A Sequence Diagram
     * actor myActor
     * database myDb
     * activate myActor
     * myActor ->> myDb: async request
     * activate myDb
     * myActor //-- myDb: response
     * destroy myDb
     * deactivate myActor
     * @enduml
     *
     * @param paramA a sample parameter
     * @return paramA % 2 == true: param is even; false: param is odd
     */
    bool MethodA(int paramA)
    {
        return paramA % 2 ? false : true;
    }
};
Word Macros

- Download Word template 'PlantUML_Template_v30.dotm'
- ! current template not suitable for 64bit version of Word !
- Copy *.dotm file to %appdata%\Microsoft\Word\STARTUP
- Note: run 'echo %appdata%' to check value of 'appdata'
- restart Word -> menu item ‘PlantUML’ should appear
Word Macros

- PlantUML script within Word document; can be hidden for final doc

```plantuml
@startuml
actor myActor
database "My Superfast Database" as myDb
note right of myDb: in the script we use the abbreviation automember
alt Synchronous messages
myActor -> myDb: sync request
else Another synchronous messages
myActor --> #black: myDb: another sync request
else Asynchronous messages
myActor -- #red: myDb: async request
myActor: //#green: myDb: response
note right: this is the response to the async request
end
caption Different arrow types
@enduml
```

A Sequence Diagram

- Different arrow types
- myActor
- My Superfast Database
- 1 sync request
- 2 another sync request
- 3 async request
- 4 response
- this is the response to the async request
Resources and Links

- PlantUML: http://plantuml.com/
- Word Template: https://sourceforge.net/projects/plantuml/files/PlantUML_Template_v30.dotm/download
- Graphviz (for dot.exe): http://www.graphviz.org/
- Doxygen: www.doxygen.org/
- Website for editing and testing scripts: http://www.planttext.com
The End

Presenter

Thank you for your attention

Questions ???

Presenter

Audience

Audience