Consistent architecture diagrams for C++ projects

Marius Feilhauer
About me

◦ M.Sc. Automotive Engineering @ University Stuttgart

◦ PhD thesis *Simulation-based validation of advanced driver assistance systems* @ HLRS University Stuttgart (Prof. Resch) and ETAS GmbH

◦ Working experience @ Robert Bosch GmbH, Aston Martin Ltd. and Porsche AG

◦ Software developer of C++ based simulation models @ ETAS GmbH

◦ Lectures @ University Stuttgart
Introduction and Overview

- Source Code
- Test
- Build Scripts
- Documentation
- Diagrams

Repository:
- v1.0
- v1.1
- v1.2
- v2.0

Build Tooling:
- v1.0
- v1.1
- v1.2
- v2.0

Download Server
Agenda

○ Good documentation

○ Architecture and code visualization

○ Consistency

○ Diagrams as code

○ Takeaways & Outlook
Good Documentation
Golden nuggets
Guiding principles by atlassian

Keep it brief

Visually are key

Know your audience
Documentation use cases

Two examples

- Mh, undocumented functions
- Cool new library

I get this issue
ah, we changed this ...
unprofessional

Adam Scott, www.oreilly.com/ideas/the-eight-rules-of-good-documentation
The eight rules of good documentation

1. Write documentation that is inviting and clear
2. Write documentation that is comprehensive, detailing all aspects of the project
3. Write documentation that is skimmable
4. Write documentation that offers examples of how to use the software
5. Write documentation that has repetition, when useful
6. Write documentation that is up-to-date
7. Write documentation that is easy to contribute to
8. Write documentation that is easy to find
Additional benefits when using consistent diagrams within your documentation

- Fast understanding of your intentions
- Professional public image
- Enforces to express intentions precisely
Diagrams and code visualization
UML Diagrams

Overview

- UML Diagram
  - Structure Diagram
    - Class Diagram
    - Object Diagram
    - Package Diagram
    - Composite Structure Diagram
    - Component Diagram
    - Deployment Diagram
    - Profile Diagram
  - Behavior Diagram
    - Use Case Diagram
    - Activity Diagram
    - State Machine Diagram
    - Interaction Diagram
      - Sequence Diagram
      - Communication Diagram
      - Timing Diagram
      - Interaction Overview Diagram

Based on www.uml-diagrams.org/uml-25-diagrams.html
Grade System Example

Use Case Diagram

Madeline Adair, Use Case Diagrams, www.slideplayer.com/slide/677748
Shopping Example

UML Component Diagram Example

Order -> Product (Item Code)
Order -> Customer (Customer Details)
Order -> Account (Payment)
Abstract Factory Design Pattern

UML Class Diagram Example
Let’s get a coffee

Activity Diagram

1. Want some coffee
2. Walk to the coffee machine
3. Brew coffee
   - want milk
     - Add milk
     - don’t like milk
       - Add milk
4. Take coffee
   - want some sugar
     - Add sugar
   - no sugar for me
5. Enjoy coffee
Smart Home Example

Deployment Diagram
Diagrams allow to transport information easily and there are many different options available how to draw them.
Consistency
Consistency

Documentation v1.0

Documentation v1.1

Class Diagram

Shape

draw()

Rectangle
draw()

Circle
draw()

Meeting C++ example

Class Diagram

BaseShape
draw()

Rectangle
draw()

Circle
draw()

Triangle
draw()

Meeting C++ example
Draw diagrams by using code

Source code of diagrams has to be kept in-sync with the rest of the code and its documentations.
Diagrams as code
Overview
Diagrams as code

- Available tools allow automated generation of diagrams
  Doxygen, CMake, PlantUML

- Easy integration into existing build toolchains

- Automatic approaches vs. scripted diagrams
Example project
Directory structure

- Example
  - Vehicle
  - Traffic Light
  - Vehicle Garage
  - Vehicle Creator
  - Race Game
  - CMake
  - Documentation
    - DependencyGraph
Example project

CMake targets

Executable
Race Game

Executable
Vehicle Creator

Dynamic Library
Traffic Light

Static Library
Vehicle Garage

Dynamic Library
Vehicle
Dependency Graphs

Cmake & Graphviz

CMakeList.txt → CMake → Graph.dot → Graphviz → Diagram

Build Tooling

https://www.graphviz.org/
Dependency Graphs
Cmake & Graphviz

```cmake
add_custom_target(DependencyGraph
    COMMAND cmake --graphviz=DependencyGraph.dot .
    COMMAND dot -Tsvg DependencyGraph.dot -o DependencyGraph.svg
    WORKING_DIRECTORY ${CMAKE_BINARY_DIR})
```

CMakeLists.txt
Collaboration Diagrams

Doxygen

```
example::Vehicle

+ setMaximumSpeed()
+ setAcceleration()
+ position()
```

COLLABORATION_GRAPH = YES
Include Graphs

Doxygen

```
INCLUDE_GRAPH = YES
INCLUDED_BY_GRAPH = YES
```
UML Diagrams

PlantUML

Diagram.plantuml

PlantUML

Build Tooling

Diagram
PlantUML

Some Examples
PlantUML

Live Demo in Visual Studio Code

Visual Studio Code & PlantUML Extension
# This is a vehicle model

This example shows a PlantUML diagram within a Markdown file to be processed by doxygen.

```plantuml
@startuml "Vehicle Simulation"

<table>
<thead>
<tr>
<th>Create Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query position</td>
</tr>
</tbody>
</table>

repeat while (stop simulation?)

stop

@enduml
```
PlantUML

Consistency

DiagramA.plantuml
DiagramB.plantuml
DiagramC.plantuml

PlantUML

Doxygen & PlantUML consistency demo
Great tooling for drawing diagrams as code is already available!
and it even integrates nicely with CMake and Doxygen
Takeaways & Outlook
Takeaways & Outlook

- Visuals are key
- Diagrams as code

- Great tooling available
- Single source & version controlled
- Fun & easy to work with

- Doxygen name resolution
- Reference of header files

Doxygen link demo
Thank you