



C Dependency Graph

Sprint 3 Demo

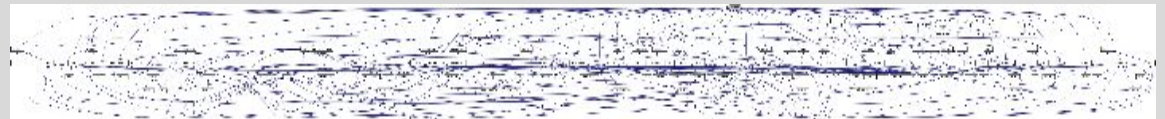
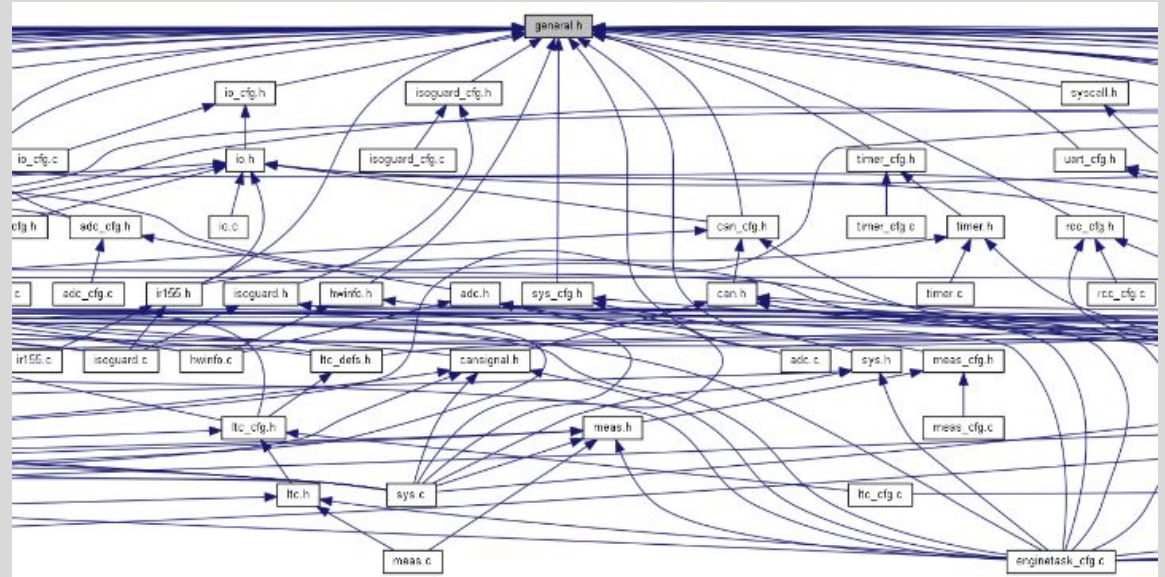
Overview of the Product:

The Problem

Illustrates
dependencies in C
code

Looks like spaghetti

This is the problem
that we set out to
fix/improve

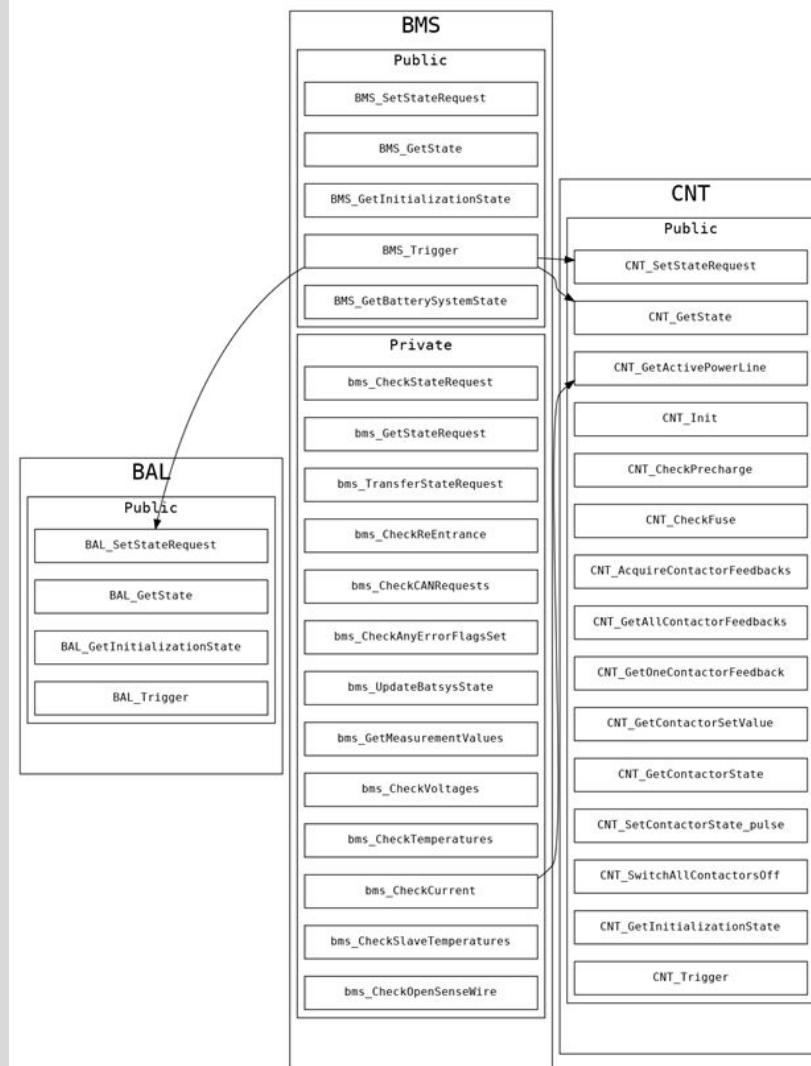


Overview of the Product: The Solution

Our objective was to “clean up” the graph

This is a concept of how our output should look

Graph should be readable and organized



Overview of the Product: The Result

Graph generated by our program

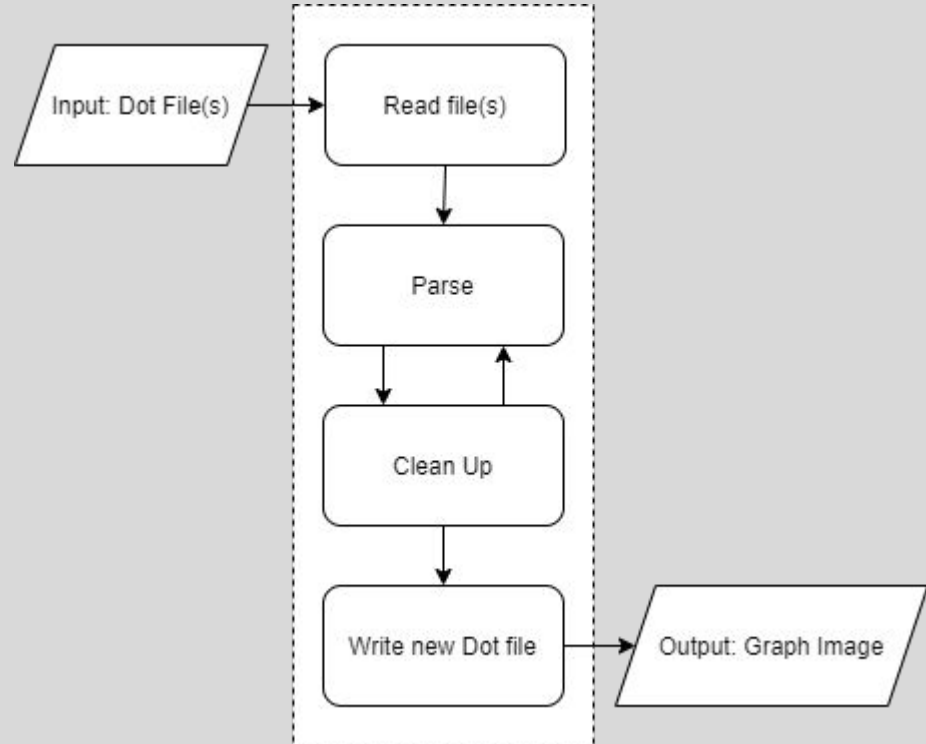


Overview of the Product: The Program

Our program takes the original graph's text files as input

We parse the files, change things about the graph, and rewrite the file

The project outputs the new graph





Overview of the Product: The Demonstration

Welcome to the C Dependency Graph Tool!

```
+ depgraph depgraph -s dot-files/adc_8h_ae0b9ae6e4ef2dbf771dcc0ea30901ae2_cgraph.dot
```



Samantha Shultz' Contribution

- Learning Dot language and other tools
- Reading in File(s) or a Directory
- Handling Private vs Public Functions
- Test Cases
- Documentation
- IEEE Conference Paper



Luis Mora's Contribution

- Learning Dot language and other tools
 - Doxygen
 - Gradle
- Coding
 - Design and implementation
- Test Cases
 - Develop initial test suite
- Documentation
 - System Requirements Specification Document
 - System Design Document
- IEEE Conference Paper



Corrina Del Greco's Contribution

- Scrum Master
- Design
 - Initial design of modules
- Code
 - Framework
 - Parser
 - Formatting, Javadocs, reviewing other's pull requests
- Testing
 - JUnit
- Documentation
 - Wiki: Guides and design
 - Test Plan: Majority, test cases, tracing to SRS
 - SDD: Design, diagrams
 - SRS: design, revising requirements



Aaron Van De Brook's Contribution

- Set-up/manage Gradle (build system) and Git (version control)
- Major contributions to overall design
- Created first manually generated graph
- Created first version of the “Lexer” module
- Responsible for “Graph Writer” and “Image Rendering” modules
- Logical flow charts of most modules

Sprint Review & Retrospective: Updated Scrum Board

Dependency Graph Scrum Board

Updated 8 days ago

Filter cards

+ Add cards(2 new)

Fullscreen



0

Product Backlog

+ ...

Automated as To do Manage

0

Sprint Backlog

+ ...

1

In Progress

+ ...

🕒

Test Parser

...

#44 opened by corrinadelgreco

🔗 Sprint 3

2

Review

+ ...

🕒

Module Filtering Functionality

...

📄 8 of 8

#70 opened by AVanDeBrook

🔗 Sprint 3

1 linked pull request

🕒

Draft IEEE Conference Paper

...

#57 opened by AVanDeBrook

🔗 Sprint 3

37

Done

+ ...

🕒

Create Class to Run DOT on Output File

...

#34 opened by AVanDeBrook

🔗 Sprint 3

2 linked pull requests

🕒

BUG: Extra node in the graph that should not be there

...

#71 opened by AVanDeBrook

🔗 Sprint 3

1 linked pull request

🕒

SRS V3 - Finalize Software Requirements Specification

...

📄 5 of 5

#65 opened by corrinadelgreco

🔗 Sprint 3

🕒

SDD V3 - Finalize Software Design Document

...

📄 5 of 5

#66 opened by corrinadelgreco


🔗 Sprint 3

🕒

TP V2 - Finalize Test Plan

...

Automated as Done Manage



Sprint Review & Retrospective: Spring Term

- Create Graphical User Interface (GUI)
- More checks for bad input/user error
- Port to Python or Ruby
- Improve output graph
- Other possible project ideas

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is light green. They are positioned diagonally, with the blue one partially covering the green one.

Thank you

C Dependency Graph