

IOWA STATE UNIVERSITY

Software Engineering

Engine Data Analysis Tool

sdmay20-06

Zachary Frisvold, Will Sartin, Thomas Haddy, John Powen, Ryan Radomski, Jamie Raught

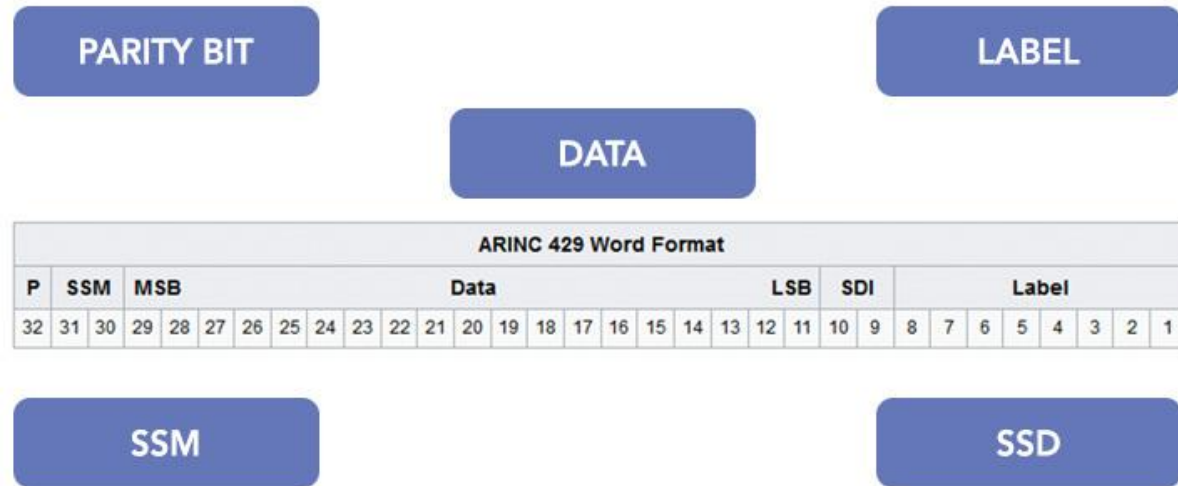
Team Site: <http://sdmay20-06.sd.ece.iastate.edu/>

Overview

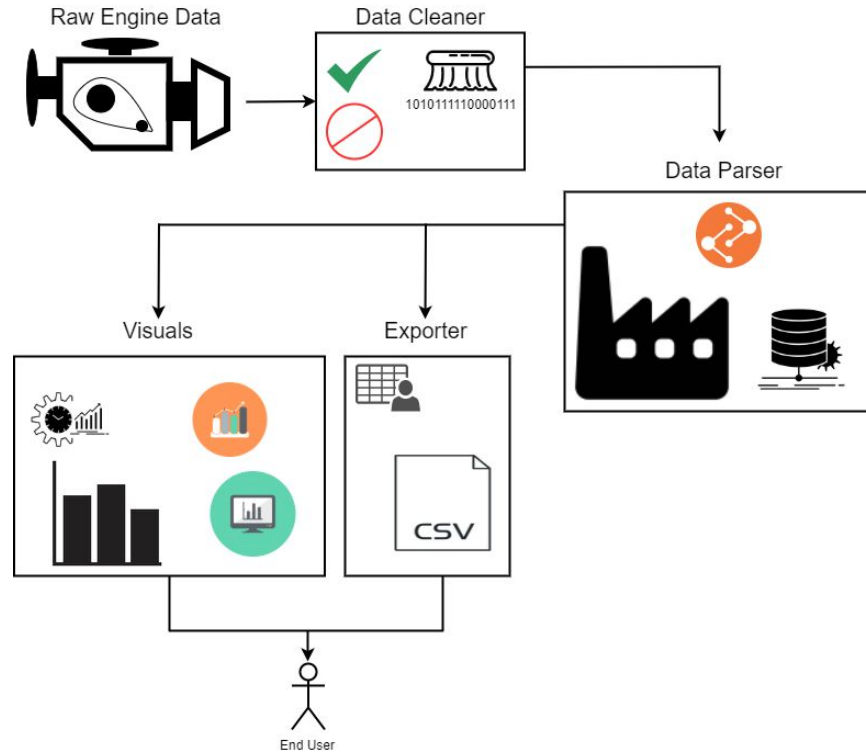
- Windows Application written in C#
- Software product for parsing and analyzing ARINC 429 data
 - Interactive graphs to analyze data
 - Export data into multiple formats

Problem Statement

- To create a program to parse a .csv log file from the center cluster of a C-130 Airplane and show the data in a user-friendly format.



Conceptual Sketch



Functional Requirements

- Windows compatible executable file
- No internet connection necessary
- Capability to create user defined graphs, charts, and tables
- Convert the ASCII data into ARINC-429
- Parse the input file into selectable data fields
- Report anomalies and provide quick access to visualizations
- Export parsed data to either .csv or .xls
- Allow playback of trend data once imported

Non-Functional Requirements

- Parsing needs to run quickly, with a worst case of 10 minutes
- Accept any .csv and .xls files as input
- Recognize if the .csv input file is valid and provide a notification on failure
- Provide data field names for predefined data as per import data
- Capability for user defined data field names
- Allow filtering what's included in the data set export
- Playback sequences the data points at 1 Hz update rate
- Allow a graphical depiction of the EIDS dials during playback

Technical Considerations

- Runtime - Quickly parse data
- Implementation
 - Windows - No other port planned
 - Dynamic parsing capabilities

What Makes This Project Unique?

- Existing Collins Product:
 - MATLAB - Takes hours to parse data files
 - Not used because of runtime
- Our Product:
 - C# - Built from bottom-up
 - Efficient runtime
 - Graphing capabilities

Risks and Mitigation

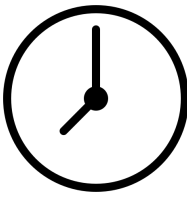
- Feature Creep
 - Timeboxing
 - Prioritization of Requirements
- Inaccurate Data Specification
- No Test Oracle

Cost Estimate

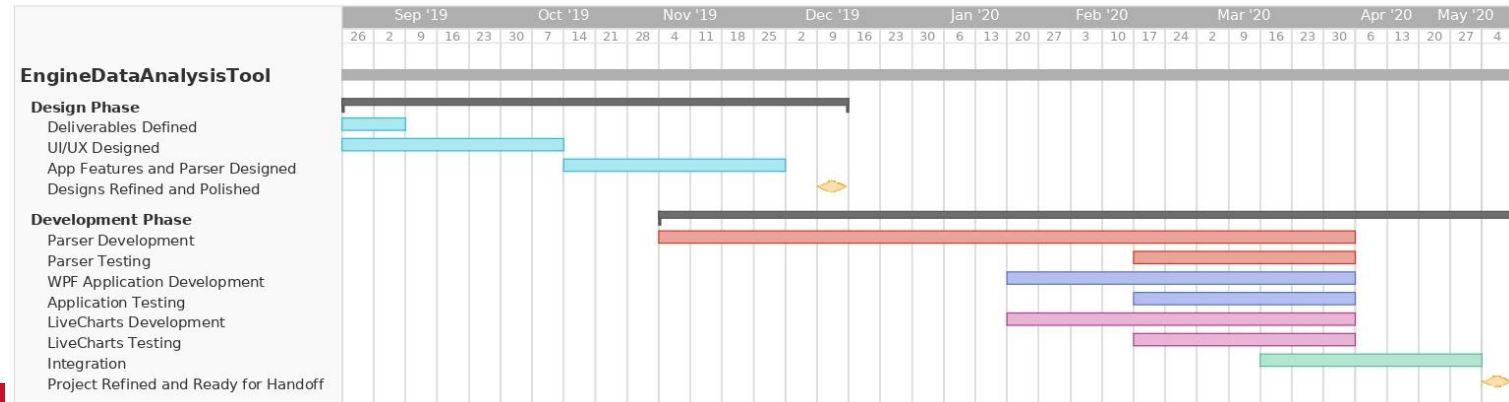
- No Budget, but Time is Money
- Create data parser in 1 semester
- Create front end component in 1 semester
 - Agile development practices – will keep adding features through entire semester



Project Milestones

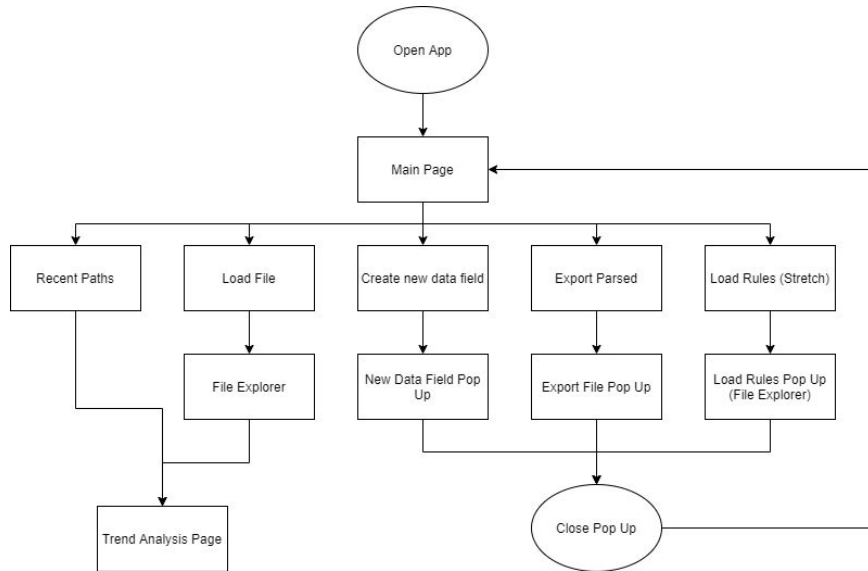


- Parser Prototype (Nov '19)
- LiveCharts Implementation (Feb '20)
- WPF Application (Mar '20)
- Data Parser (Mar '20)
- Data Exporter (Apr '20)
- Integration (Apr '20)

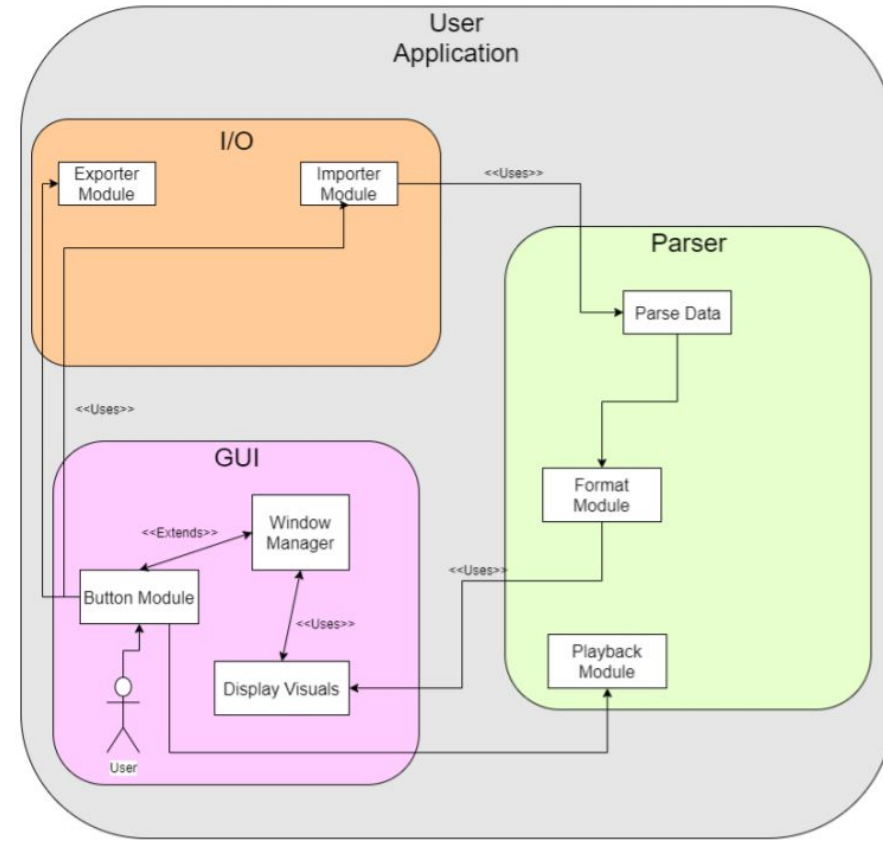


Design

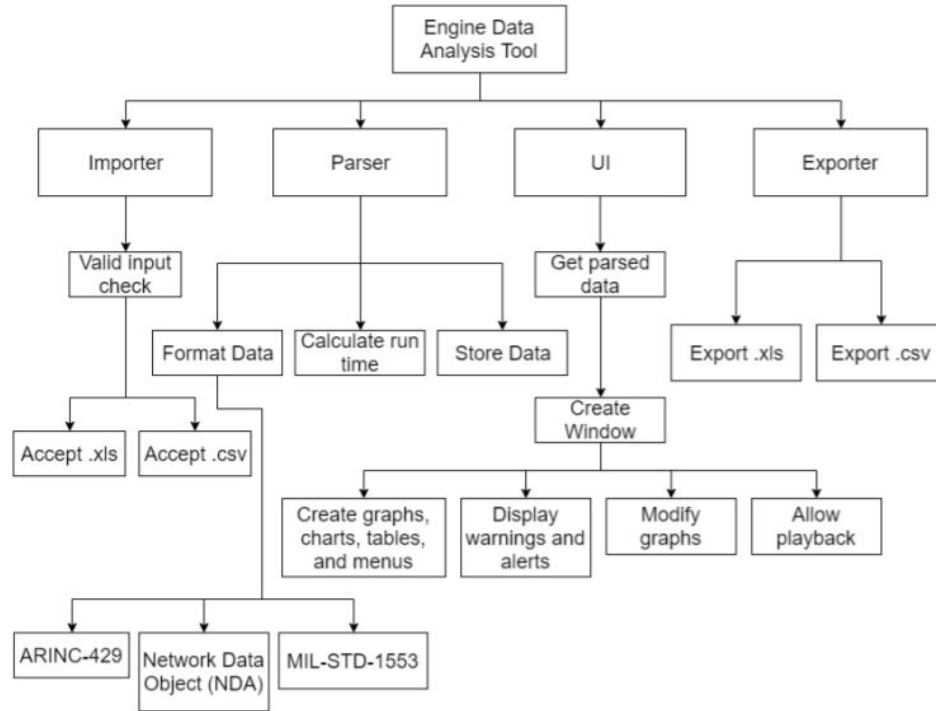
- Architecture: Model View Viewmodel



System Block Diagram



Functional Decomposition



Technology Platforms Used

- Language - C#
- .NET Framework 4.7.2
- Library - LiveCharts for graphing UI
- WPF - Platform for Windows Desktop apps
- IDE - Visual Studio



Test Plan

- Test-as-you-go strategy
- Regression testing on new features
- Sections:
 - GUI data-handling and UI
 - Parsing capability
 - Exporting functions
 - Example data with LiveCharts

Prototype Implementation

- Data Parser from spec
- Data Exporter to selected format
- WPF Application
- Interactive Graphing

ARINC 429 Word Format																																	
P		SSM			MSB																	Data		LSB		SDI		LSB		Label		MSB	
32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		

<?xml version="1.0" encod

- ▼ plane [34]
 - ▶ power_on_counter [1]
 - ▶ fuel_quantity_totalizer [1]
 - ▶ fuel_quantity_outboard [2]
 - ▶ fuel_quantity_inboard [2]
 - ▶ fuel_quantity_aux [2]
 - ▶ fuel_quantity_ext_inboard [2]
 - ▼ torque [4]
 - ▶ engine_1 [1]
 - ▼ engine_2 [1]
 - ▶ ARINC_429_Word [5]
 - ▶ engine_3 [1]
 - ▶ engine_4 [1]
 - ▶ turbine_inlet_temperature [4]
 - ▶ engine_oil_pressure [4]
 - ▶ engine_oil_temperature [4]

Engineering Standards & Design Practices

- **IEEE 1008-1987** Standard for Software Unit Testing
- Principle of Least Surprise/Astonishment
 - Naming conventions, WorkFlow.
- Principle of Least Knowledge
 - Modularity & minimizing dependencies



Team Contributions

| | |
|---------|----------------------------|
| Zak: | Communication Lead |
| Will: | Meeting Facilitator |
| Thomas: | UI/UX Architect |
| Ryan: | Quality Assurance engineer |
| John: | Scrum master |
| Jamie: | Tech Owner |

Future Prospects

- Extend exporting capabilities
- Create User Guide
- Hand-off source code & executable

