**Project Specification Document**

Adventium Labs

Ansley‌ ‌Schug‌ ‌

Charles‌ ‌Kanoy‌ ‌

Jeremy‌ ‌Rylicki‌ ‌

Riley‌ ‌Abrahamson‌

**Table of Contents**

Section Page

[1.](#_heading=h.30j0zll) Project Vision and Objectives 1

[1.1](#_heading=h.1fob9te) Project Scope and Vision 1

[1.2](#_heading=h.3znysh7) Project Goals and Objectives 1

[2.](#_heading=h.2et92p0) Project Tracking 2

[2.1](#_heading=h.tyjcwt) Deliverables 2

[2.2](#_heading=h.3dy6vkm) Project Metrics/Project Acceptance Criteria 2

[3.](#_heading=h.4d34og8) Project Closure 3

[3.1](#_heading=h.2s8eyo1) Delivered Solution 3

[3.2](#_heading=h.17dp8vu) Remaining Work 3

[3.3](#_heading=h.3rdcrjn) Future Work Suggestion 3

[4.](#_heading=h.26in1rg) Definitions and Acronyms 4

# 1. Project Vision and Objectives

## 1.1 Project Scope and Vision

As a small portion of the GUMBO project, our group is responsible for creating an OSATE plug-in that extends the AADL models generated from FACE diagrams with AGREE annexes. The vision for our project is to make a simple user interface plugin for the Eclipse IDE. This interface will have a selection experience to reduce the likelihood of error in the generation of AGREE constraints. The end user, members of the US Army, will not have a lot of experience with actually programming in AADL, so this project is critical to creating an error-free final AGREE model with proper constraints.

## 1.2 Project Goals and Objectives

|  |  |
| --- | --- |
| **#** | **Goal or Objective** |
| 1 | Create an OSATE plugin that: |
| 2 | * Reads an AADL instance models through the OSATE API |
| 3 | * Identifies components translated from FACE |
| 4 | * Create GUI that opens a dialog box that collects behavioral details on these components |
| 5 | * Output results as AGREE annex that defines interface behaviors |

# 2. Project Tracking

## 2.1 Deliverables

|  |  |
| --- | --- |
| **#** | **Deliverable** |
| 1 | Source code (Java) |
| 2 | README including installation and running instructions |
| 3 | Demo video |
| 4 | Project specification |
| 5 | Post-Mortem document |
| 6 | Final report |

## 2.2 Project Metrics/Project Acceptance Criteria

|  |  |  |
| --- | --- | --- |
| **Metric** | **Frequency** | **Location** |
| User Story Points from Monthly Report Feedback | End of each month | At the beginning of Individual Sprint (Section 4.5) |

# 3. Project Closure

## 3.1 Delivered Solution

Currently, the solution has independent front and backends. The UI for the frontend is satisfactory and fulfills the requirements specified from the beginning of the project. The user can create new assumptions and guarantees and generate a copy-pastable output to extend their models. The backend correctly iterates through a selected model and grabs nearly all of the necessary data to create extensions in the front end.

## 3.2 Remaining Work

There are a few bits of improvements to be made across the project.

* Connecting the front and backends and shift the UI off of sample data
* Extend the front-end comparator drop-down menus to accommodate different data types (ex. boolean vs number data types)
* Within the backend iteration, pull the data type values from the specific parameters for inputs and outputs
* Create a data persistence model to allow a user to save their work between annex generations
* Package up the plugin and bundle it for development installs.

## 3.3 Future Work Suggestion

The stretch goal that would’ve been next in the priority list would be the ability for the plugin to directly edit model files and insert the generated AGREE annexes into the model code. This would save the user from having to copy-paste the data they generate. This was specified by Adventium Labs to be a bonus, and not necessary to put the plugin into production environments for employees.

# 4. Definitions and Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| FACE | **Future Airborne Capabilities Environment:** Model used to describe system architectures |
| AGREE | **Assume Guarantee Reasoning Environment:** Used to model assumptions and guarantees for AADL models |
| OSATE | **Open Source AADL Tool Environment:** A Eclipse-based IDE for AADL modeling |
| AADL | **Architecture Analysis and Design Language:** Used for modeling software and hardware systems |
| GUMBO | **Grand Unified Modeling of Behavioral Operators:** Overarching project aiming to unify cross-section of behavior representations in AADL to support unified behavioral analysis. |