Requirements Document

Project Title: Rules Based Decision Aid Framework

Distribution:

ASRC Federal Mission Solutions, Kimberly Davis

ASRC Federal Mission Solutions, Kevin Wainwright

ASRC Federal Mission Solutions, Christopher Barone

Rowan University, Professor Jack Myers

Table of Contents

1 Purpose …………………………………………………………………………………………………… 2

2 Scope ……………………………………………………………………………………………………... 2

3 Exclusions, Assumptions, and Limitations ………………………………………………………………… 3

3.1 Exclusions ………………………………………………………………………………………….. 3

3.2 Assumptions ……………………………………………………………………………………….. 3

3.3 Limitations ………………………………………………………………………………………. 3

4 Requirements ……………………………………………………………………………………... 5

4.1 Non-Functional Requirements …………………………………………………………………….. 5

4.2 Functional Requirements ………………………………………………………………………... 6

4 Roles and Responsibilities………………………………………………………………………………. 7

5 Terms and Definitions ………………………………………………………………………………. 7

1. Purpose

The purpose of this requirements document is to provide documentation. This document records User, Solution, and Compliance requirements for Rules Based Decision Aid Framework. This document will be maintained throughout the life of the in-scope Solution Configuration Items.

2 Scope

Project Title: Rules Based Decision Aid Framework

Project Justification:

We are to automate the process of developing accurate courses of action based on sensory data of a C2 system by using a RBDE. Having C2 system pass data through a RBDE will make it so that decisions can be made in a more timely and efficient manner by the operators using it. It will also make changing and extending the system's decision models much easier.

Project Scope:

This project will consist of creating a rules based decision engine to further augment a C2 system. The project will include various installments including importing data from an external source, running that data through rules and creating goals from this data. It will also have functionality for the user to create new rules as necessary and to chain rules off of other rules. Although not as important as the previously stated installments, the project should also include the capability to turn on and off different rules, search for specific rules and have a basic GUI for user control.

Project Deliverables:

- *Bi-weekly functional installments of the project*
- *Proper project functionality*
- *UML Diagram*
- *Use Case Diagram*
- *Readme/User Guide*
- *Summary of Changes*
- *Burnup/Burndown Charts*
- *Pre-Sprint Goals*
- *Testing for the Project*

Project Success:

The project will be determined successful if it is a fully functional project that meets not only all of the user requirements, but MSE definition of done as well.

3 Exclusions, Assumptions, and Limitations

3.1 Exclusions

Development of a proxy that would enable us to pipeline data from an onboard sensor directly into the RBDE system was strictly excluded from the prototype development.

3.2 Assumptions

It is assumed that users of the RBDE system will be subject matter experts in their specific field. Users of the system are understood to have the ability to create specific rules based off of the imported data set. It is also assumed that data entering the RBDE will be one of the supported file types. To increase access to previously created rules, it is assumed that there will be some type of persistent storage system used to store data.

3.3 Limitations

The RBDE system prototype is limited to a three month development cycle, having an expected completion date in December 2016. The project budget has a limit of $0, forcing us to accept only open source software and effectively giving us no room to explore alternative COTS software. The supported data formats are limited to JSON, CSV, and standard text files. Software development skills also proved to be a development limitation as the bulk of development was forced to be completed in Java.

4 Requirements

This section of the document will discuss the Requirements of the RBDE system.

4.1 Non-Functional Requirements

Req #	***NON-FUNCTIONAL REQUIREMENTS***
1	*The RBDE system shall be compatible with Windows, Unix, and Linux platforms.*
2	*The underlying rules engine shall be open-source and cost free.*
3	*The RBDE shall accept JSON, CSV, and proprietary text file configurations.*
4	*There shall only be one instance of the RBDE system.*
5	*The RBDE shall accept only one data input source at any given time.*
6	*Only the current user should not be able to modify rules while the RBDE system is in operation.*
7	*The RBDE system should be able to run at least 10 rules simultaneously.*
8	*The user shall be able to ADD, REMOVE, or MODIFY a rule during runtime.*
9	*The user shall be able to ADD, REMOVE, or MODIFY an action during runtime.*
10	*The user shall be able to CHANGE an incoming data input source during runtime.*
11	*The RBDE system shall allow for rules to be chained and unchained.*
12	*User created rules shall be stored in persistent storage.*
13	*The user shall be able to see what rules are being activated during runtime.*
14	*The user of the RBDE system should be able to configure an input source and a set of rules within 5 minutes.*

4.2 Functional Requirements

Req #	***FUNCTIONAL REQUIREMENTS***
15	*When operating the RBDE system, the user shall be able to add an input data source file directly from the GUI.*
15.1	*When pressing the "Add Data" button the system shall allow user to browse the file system for an input data source.*
15.2	*Once the data source is selected the RBDE system shall verify that the input file is an accepted file type.*
15.3	*If the data input source isn't the correct file type, the system will notify the user of this error.*
16	*When the user presses the "Import Data" button, the RBDE system shall parse the file and store its data.*
17	*When pressing the "Remove Rule" button for a selected rule in the rule set, the RBDE System shall remove that rule from the list of active rules.*
18	*The user shall be able to create a new rule.*
18.1	*When pressing the "Create Rule" button on the GUI, a separate GUI shall pop up allowing the user to enter data about the rule.*
18.2	*The newly entered data shall be checked for correctness prior to the rule being generated.*
19	*When the user presses the "Create Rule Set" button the RBDE system shall allow the user to populate a list with readily available rules.*
20	*From the GUI, there shall be a selectable option for each rule to allow the rules to be chained together.*
21	*When viewing the list of rules in the rule set from the GUI, the user shall be able to activate and deactivate rules individually by pressing a button located next to the rule.*
22	*The GUI shall have a button next to the rule allowing user to edit the rule.*
23	*From the GUI, the user shall be able to see a list of created rules for the current data set.*
24	*The GUI shall have a button next to a created rule that allows the user to delete the rule when pressed.*

5 Roles and Responsibilities

Role	*Responsibilities*
Klaydon Balicanta	Scrum Master: enforces scrum principles and rules and assists the development team if they need assistance
Michael Moscariello	Product Owner: keeps in contact with the MSE customer if the development team has any questions and in charge of adding PBI's into the product backlog
Ian Markind Trae Lewis Michael Crinite Shiv Patel	Development Team Member: Does research accordingly and creates a fully functional product which adheres to the customer's' version of Done

6 Terms and Definitions

Below is a comprehensive table listing all acronyms listed throughout this document.

Term or Acronym	*Definition*
RBDE	*Rules Based Decision Engine*
JSON	*JavaScript Object Notation*
C2	*Command and Control*
MSE	*Mission Solutions Engineering*
COTS	*Commercial off-the-shelf*
PBI	*Product Backlog Item*

7