Hazard Analysis
MTOBridge

Team 15, Alpha Software Solutions
Badawy, Adham
Yazdinia, Pedram
Jandric, David
Vakili, Farzad
Vezina, Victor
Chiu, Darren

Table 1: Revision History

Date	**Developer(s)**	**Change**
October 12 2022	Darren	Added System Boundaries & Components
October 19 2022	Adham	Added Adham/Victor/Farzads FMEA work into a latex table
October 19 2022	Pedram	Added Safety reqs and Roadmap
October 19 2022	Victor	Changes to FMEA table formatting
April 04 2023	Victor	Revision 1

Contents

List of Tables

1 Introduction

This document is a hazard analysis of MTOBridge. Background for the project can be found in the Problem Statement. A hazard is a potentially harmful event resulting from the conditions of the system and the environment.

2 Scope and Purpose of Hazard Analysis

This document describes the components of MTOBridge, the potential hazards associated with each, and any new functional requirements that can be derived from these hazards. This process is important to identify any potential issues with the system, and then design the system to eliminate the potential issues.

3 System Boundaries and Components

This hazard analysis addresses the system that consists of the following components:

1. UI Component, for providing a graphic display to the user and visualizing MATLAB results
2. Input Handler Component, for processing user inputs
3. MATLAB Interaction Component, for calling scripts and supplying specified arguments to them
4. MATLAB Engine Component, for performing bridge calculations
5. File Manager Component, for reading inputs from files and saving results in various formats

The system boundary includes these software components and any dependency files required for the application to operate. Although the MATLAB Engine Component is owned by the client and its exact contents verified independent of this project, this hazard analysis will address it due to being a crucial component of the system.

4 Critical Assumptions

We will not be making any critical assumptions about the system. As mentioned above, the MATLAB engine is a critical component of the system. We must address it with the hazard analysis and will not make any assumptions about its correctness and functionality.

5 Failure Mode and Effect Analysis

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI displays truck config incorrectly.	User confusion / misleading interface.	a. Failure to catch invalid user input. b. Bug or error in truck display module. c. Unexpected/boundary case user input.	a. Ensure truck display module has correct input bounds and safety net to catch invalid user input. b. Thoroughly test the truck display module to avoid unexpected responses to input. c. User input modules will be tested thoroughly to ensure that all boundary cases are accounted for.	None	HA-1
UI	UI does not update to match truck config.	User missing important information.	a. Failure to catch invalid user input. b. Bug or error in truck display module. c. Unexpected/boundary case user input.	a. Same as HA-1a b. Same as HA-1b c. Same as HA-1c	None	HA-2

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI displays bridge config incorrectly.	User confusion / misleading interface.	a. Failure to catch invalid user input. b. Bug or error in bridge display module. c. Unexpected/boundary case user input.	a. Ensure bridge display module has correct input bounds and safety net to catch invalid user input. b. Thoroughly test the bridge display module to avoid unexpected responses to input. c. Same as HA-1c.	None	HA-3
UI	UI does not update to match new bridge config.	User missing important information.	a. Failure to catch invalid user input. b. Bug or error in bridge display module. c. Unexpected/boundary case user input.	a. Same as HA-3a. b. Same as HA-3b. c. Same as HA-1c.	None	HA-4

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI attempts to display undesired calculation type.	Display is incorrect.	a. Incorrect processing of user input. b. Incorrect/misleading display of user solver selection.	a. Thoroughly test solver configuration module ensure correct processing of user input. b. Minimize complexity of input handler/solver selection display modules and their interaction to reduce chances of incorrect information being shared.	None	HA-5
UI	Truck platoon animation does not match bridge load animation.	Display is difficult for user to parse.	a. Incorrect calculation in display logic. b. Unexpected bug or glitch in calculation display modules.	a. Thoroughly test calculation display module(s) to avoid unexpected behavior. b. Include check of truck display location (as in coordinates of trucks in the animation) vs bridge load animation frame within display logic to endure the two displays are alligned. Re-calculate and restart animation if they are not.	SR-1	HA-6

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	Platoon trip and bridge load sync check(s) provide false positives or negatives.	Deny a fine display or let through an erroneous one.	a. Incorrect sync check logic. b. Unexpected bug or glitch in sync module.	a. Thoroughly test sync checks to avoid unexpected behavior. b. Simplify sync check logic while maintaining accuracy to limit the chance of incorrect logic.	SR-1	HA-7
UI	UI incorrectly displays the concerned section.	Display is incorrect.	a. Unexpected bug/glitch in concerned section display module. b. Misleading or incorrect display of user concerned section selection.	a. Thoroughly test concerned section display module to ensure correct processing of user input. b. Minimize complexity of input handler/concerned section display modules and their interaction to reduce chances of incorrect information being shared.	None	HA-8

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI incorrectly displays discretized bridge segments.	Display is incorrect.	a. Unexpected bug/glitch in concerned section display module. b. Misleading or incorrect display of user concerned section selection.	a. Same as HA-8a. b. Same as HA-8b.	None	HA-9

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI fails to display calculation results entirely.	Display is incorrect.	a. Unexpected/boundary case user input. b. Runtime error in calculation display modules. c. Failure to catch invalid user input.	a. Thoroughly test the calculation display modules to avoid unexpected responses to input. b. Ensure calculation display modules have robust error handling to avoid catastrophic failure if an error is encountered, and design modules with separation of concerns in mind to limit error propagation. c. Ensure calculation display modules have correct input bounds and safety net to catch invalid user input.	None	HA-10
UI	UI encounters parallel computing issue such as deadlock/race condition.	Incorrect results or unexpected program behavior	a. Multiple threads modifying the same values/waiting on each other.	a. Implement proper thread safety measures to avoid deadlocks, race conditions, etc...	SR-2	HA-11

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
UI	UI stops reacting to user inputs	User locked out from using UI, program is unusable.	a. Runtime error in calculation display modules. b. Parallel computing issue such as deadlock that hangs the program.	a. Ensure all display modules have robust error handling to avoid catastrophic failure if an error is encountered, and design all modules with separation of concerns in mind to limit error propagation. b. Same as HA-11a.	SR-2	HA-12

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Matlab Interaction	Data received is incorrectly formatted.	Program cannot display or save calculation results.	a. One-time error in cross-program communication caused by outside factors (OS, hardware, etc.). b. Bug or error in MATLAB engine.	a. Try all calculations a second time if first calculation fails. If it fails a second time, then the errors are most likely being produced by cause b and those recommended actions must be taken. b. The MATLAB engine will be tested thoroughly to try to reduce the amount of bugs it has. The program will always log in-depth error information and display an error message to the user telling them to contact the developers when there is an issue with the MATLAB component.	SR-3, SR-4	HB-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Matlab Interaction	Unable to call engine.	Program cannot perform calculations.	a. One-time error in cross-program communication caused by outside factors (OS, hardware, etc.). b. Engine not installed / installed improperly.	a. Same as HB-1a. b. The program will always display a message telling the user that they must install the MATLAB engine with a reference to the installation section of the user manual when the MATLAB engine is not detected.	SR-3, SR-4	HB-2
Matlab Engine	The engine crashes unexpectedly.	Program cannot display or save calculation results.	a. One-time crash caused by outside factors (OS, hardware, etc.). b. Bug or error in MATLAB engine.	a. Same as HB-1a. b. Same as HB-1b.	SR-3, SR-4	HC-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Matlab Engine	The engine calculations take more time than should be required (more than 10 seconds).	Program must wait for results.	a. One-time error causing infinite looping caused by outside factors (OS, hardware, etc.). b. Bug or error in MATLAB engine.	a. Same as HB-1a. b. Same as HB-1b.	SR-3, SR-4	HC-2
Matlab Engine	Data received from the engine is incorrect (as in physically impossible).	Program cannot display or save calculation results as they are incorrect.	a. One-time calculation error caused by outside factors (OS, hardware, etc.). b. Bug or error in MATLAB engine.	a. Same as HB-1a. b. Same as HB-1b.	SR-3, SR-4	HC-3
Input Handler	Handler passes out of bounds inputs to other components.	Inaccurate analysis results.	a. Accidental changes of input, for example writing 100,000 instead of 10,000,000.	a. Validate numeric values are within an acceptable range.	SR-5	HD-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
Input Handler	Handler passes incorrect type of input to other components.	System crash.	a. Accidental mix and match of inputs, for example inputting string into bridge length.	a. Validating input type before passing it on.	SR-6	HD-2
Input Handler	Handler passes on incomplete set of inputs.	System crash.	a. Submitting before completing all input.	a. Detecting if required inputs are missing. If they are then notify user that all inputs must be completed before calculations can be started, and do not pass on incomplete inputs.	None	HD-3
File Manager	File Manager loads corrupted configuration or saved files.	Inaccurate results or system crash.	a. Process responsible for creating the file was interrupted. b. Files edited manually by user.	a. Have metrics that indicates file creation was completed and it is communicated to the user when loading if not. b. Have metrics such as checksums to ensure the integrity of the files.	SR-7	HE-1

Table 2: FMEA Analysis

Component	**Failure**	**Effect**	**Cause**	**Recommended Action**	**SR**	**Ref**
File Manager	File Manager partially saves file.	Data loss.	a. Power outage. b. System crash.	a. Automatically save to a file whenever user changes the configuration or at reasonable time intervals. b. Same as HE-2a.	SR-8	HE-2
File Manager	File containing saved data is lost.	User cannot load previously saved data.	a. File was moved or deleted by user. b. File was deleted by the system file manager.	a. When saving a file, the user will be prompted to input a memorable and meaningful name and location on disk. Saved files will have a specific extension indicating that they are files associated with the program. b. User will be warned if the location they are saving the file to is on a disk with very limited space.	None	HE-3

6 Safety and Security Requirements

Using the results of FMEA, we can derive the following safety and security requirements for our system in order to mitigate the identified hazards.

SR-1: The system must be able to synchronize the display of the truck platoon with the bridge load. A correction will be made if the two parts are out of sync, or an error will be displayed if necessary.

Rationale: While splitting the calculation display modules into two modules, truck platoon and bridge load display, can simplify each part, it can lead to mismatch in each respective display.

Trace: HA-6, HA-7

SR-2: The system must have thread safety between the UI and other connected components.

Rationale: In order to avoid race conditions and deadlocks that could result in undesirable behavior, thread safety must be an integral part of the interaction between components.

Trace: HA-11, HA-12

SR-3: The system must produce a detailed log of function calls made to MATLAB Engine. The log will include timestamps along with software environment information such as current input.

Rationale: The logs will be directly used in the debugging process which can be presented in different ways to the user. Such logs can prove useful in case of engine crashes, data loss and timeouts.

Trace: HB-1, HB-2, HC-1, HC-2, HC-3

SR-4: In case of an Matlab engine failure, the system must be able to use the recorded logs and provide a clear message to the user.

Rationale: The logs may have too much information for a user, which may not be helpful. A simple error message should be shown to the user to inform them of the issue.

Trace: HB-1, HB-2, HC-1, HC-2, HC-3

SR-5: The system must validate numeric values are within an acceptable range before being passed on to other components.

Rationale: Accidental changes of input can cause massive shifts in the analysis and cause inaccurate results (e.g. adding extra 0's).

Trace: HD-1

SR-6: The system must validate input type before being passed on to other components.

Rationale: This is to prevent accidental mix and match of inputs.

Trace: HD-2

SR-7: The system must include information that indicates the integrity of the file and whether the file creation is completed.

Rationale: This can help remove the risks associated with corrupt files during an import or export.

Trace: HE-1

SR-8: The system must automatically save to a file whenever user changes the configuration or at reasonable time intervals.

Rationale: This is meant to mitigate risks associated with system crashes or power outages.

Trace: HE-2

7 Roadmap

The new requirements derived in the previous section can be prioritized based on time to develop, probability, and severity. We believe functionality that, if not implemented, can disrupt the user-flow must be integrated into the system. These requirements will be developed as part of the primary development phase, while the rest are allocated as stretch requirements for later implementation.

7.1 Phase 1

Critical requirements include SR-1, SR-2, SR-5 and SR-6. SR-1 is meant to mitigate the risk of mismatch and miscalculations across modules which, even though very unlikely, can cause faulty results with a low chance of detection. SR-2 is to avoid any problems that may occur in a multi-threaded system, where there is significant risk. SR-5, SR-6 are meant to mitigate the risks associated with user input. While these requirements may take a long time to develop, they have high probability and severity and should be implemented as soon as possible.

7.2 Phase 2

Phase two requirements include SR-3, SR-4, SR-7 and SR-8. SR-3 and SR-4 are to implement the logging system which will directly support debugging, especially due to engine and pipeline errors. Supporting debugging would be helpful for developement and for future changes, but it is not an important feature to have immediately. SR-7 is a security requirement meant to validate imported files and their integrity through different methods. While a fairly important requirement, we believe that it has a low probability of occurrence and high time to develop. Similarly, SR-8 is another requirement which can protect the user against unforeseen environment changes such as computer crashes or power outages. Such requirements are of low priority and are scheduled as part of the stretch requirements for later development.