

# Software Requirements Specification

## MTOBridge

Team 15, Alpha Software Solutions

Badawy, Adham

Yazdinia, Pedram

Jandric, David

Vakili, Farzad

Vezina, Victor

Chiu, Darren

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	1
1.3	Mandated Constraints . . . . .	1
1.4	Naming Conventions and Terminology . . . . .	1
1.5	Relevant Facts and Assumptions . . . . .	1
<b>2</b>	<b>Functional Requirements</b>	<b>3</b>
2.1	The Scope of the Work . . . . .	3
2.1.1	The Context of the Work . . . . .	3
2.1.2	Work Partitioning . . . . .	4
2.2	The Scope of the Product . . . . .	5
2.2.1	Product Boundary . . . . .	5
2.2.2	Individual Product Use Cases . . . . .	5
2.3	Functional Requirements . . . . .	6
2.4	Phase in Plan . . . . .	9
<b>3</b>	<b>Non-functional Requirements</b>	<b>9</b>
3.1	Look and Feel Requirements . . . . .	9
3.2	Usability and Humanity Requirements . . . . .	9
3.3	Performance Requirements . . . . .	11
3.4	Operational and Environmental Requirements . . . . .	12
3.5	Maintainability and Support Requirements . . . . .	12
3.6	Security Requirements . . . . .	12
3.7	Cultural Requirements . . . . .	13
3.8	Legal Requirements . . . . .	13
3.9	Health and Safety Requirements . . . . .	13
<b>4</b>	<b>Project Issues</b>	<b>13</b>
4.1	Open Issues . . . . .	13
4.2	Off-the-Shelf Solutions . . . . .	14
4.3	New Problems . . . . .	14
4.4	Tasks . . . . .	14

4.5	Migration to the New Product . . . . .	14
4.6	Risks . . . . .	14
4.7	Costs . . . . .	14
4.8	User Documentation and Training . . . . .	14
4.9	Waiting Room . . . . .	14
4.10	Ideas for Solutions . . . . .	15
<b>5</b>	<b>Appendix</b>	<b>17</b>
5.1	Symbolic Parameters . . . . .	17
5.2	Likely Changes . . . . .	17
5.3	Traceability Matrix . . . . .	17
5.4	Reflection . . . . .	17

Table 1: Revision History

<b>Date</b>	<b>Developer(s)</b>	<b>Change</b>
October 4	David	Add context/partitioning of work
October 4	Darren	Added some Non-functional Requirements
October 4	Victor	Project Issues
October 5	Victor	Added some Non-functional Requirements
October 5	Darren	Refined Non-functional Requirements
October 5	Pedram	Added Individual Product Usecase

## List of Tables

1	Revision History . . . . .	iii
2	Business Event List . . . . .	4

## List of Figures

1	Context Diagram of MTOBridge . . . . .	3
2	Use Case Diagram of MTOBridge . . . . .	5

This document describes the requirements for MTOBridge. The template for the Software Requirements Specification (SRS) is a subset of the Volere template [Robertson and Robertson \(2012\)](#). In addition, we have added a few sections not in the Volere template: Appendix (and its subsections), Phase in Plan.

# **1 Project Drivers**

## **1.1 The Purpose of the Project**

## **1.2 The Stakeholders**

### **1.2.1 The Client**

### **1.2.2 The Customers**

### **1.2.3 Other Stakeholders**

## **1.3 Mandated Constraints**

## **1.4 Naming Conventions and Terminology**

## **1.5 Relevant Facts and Assumptions**

User characteristics should go under assumptions.



## 2 Functional Requirements

### 2.1 The Scope of the Work

#### 2.1.1 The Context of the Work

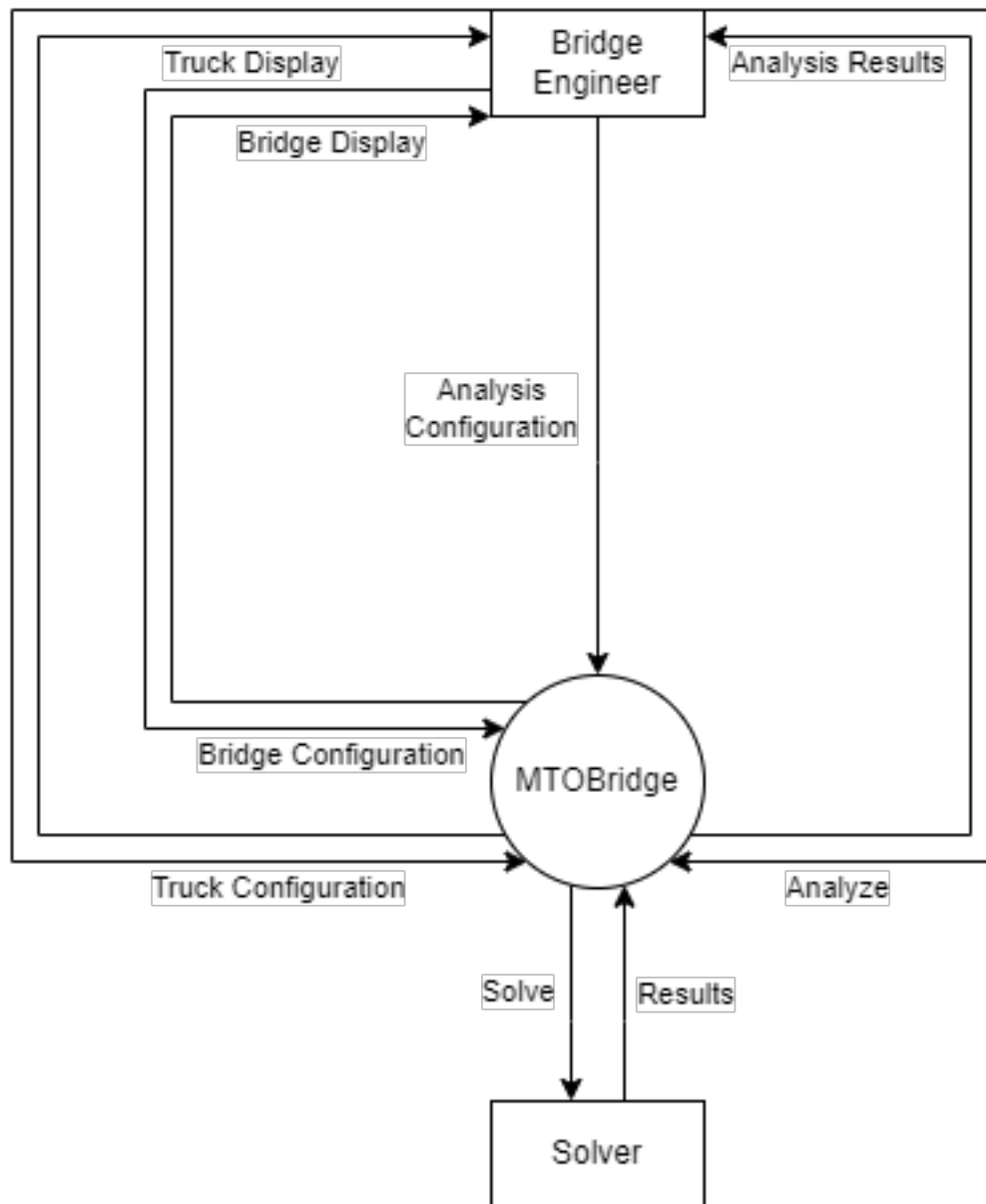


Figure 1: Context Diagram of MTOBridge



### 2.1.2 Work Partitioning

Table 2: Business Event List

Event Name	Input/Output	Summary
Engineer enters truck configuration	IN: Truck configuration, OUT: Truck display	Record truck configuration, show truck visualization
Engineer enters bridge configuration	IN: Bridge configuration, OUT: Bridge display	Record bridge configuration, show bridge visualization
Engineer enters analysis configuration	IN: Analysis configuration	Record analysis configuration
Engineer requests analysis	IN: Analyze request, OUT: Analysis results	Display analysis results
Time to solve forces	OUT: Solve request, IN: Solver results	Give configurations to solver to get results

## 2.2 The Scope of the Product

### 2.2.1 Product Boundary

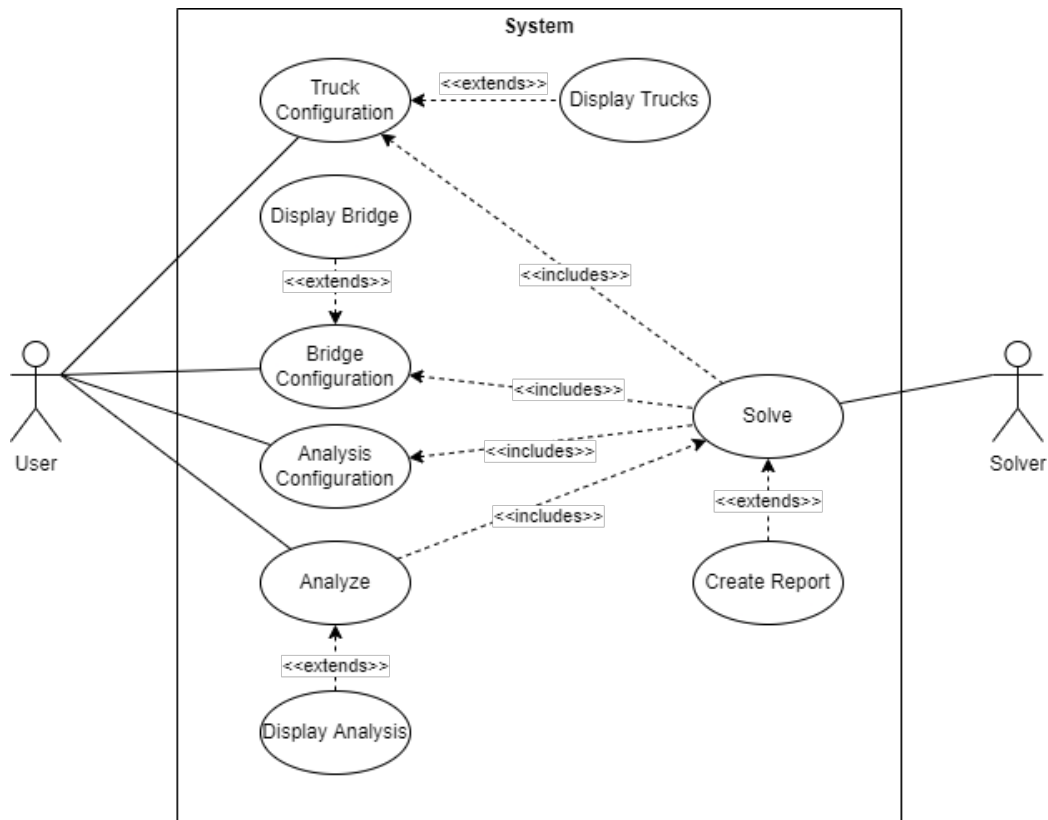


Figure 2: Use Case Diagram of MTOBridge

### 2.2.2 Individual Product Use Cases

#### User selects the Truck Configuration Tab

*Pre Condition:* None

*Post Condition:* None

*Basic Flow:* None

*Extension:* None

#### User selects the Truck Configuration Tab

*Pre Condition:* None

*Post Condition:* None

*Basic Flow:* None

*Extension:* None

#### **User selects the Truck Configuration Tab**

*Pre Condition:* None

*Post Condition:* None

*Basic Flow:* None

*Extension:* None

#### **User selects the Truck Configuration Tab**

*Pre Condition:* None

*Post Condition:* None

*Basic Flow:* None

*Extension:* None

#### **User selects the Truck Configuration Tab**

*Pre Condition:* None

*Post Condition:* None

*Basic Flow:* None

*Extension:* None

## **2.3 Functional Requirements**

**FR.1** The Program should be able to call the backend MATLAB functions.

**Rationale:** To visually display the outputs corresponding to user input, the UI needs to first figure out what those outputs are via the backend MATLAB.

**Fit Criterion:** Relatively Binary, analyze whether or not the program can successfully call the MATLAB functions.

**FR.2** The Program should allow the user to define the characteristics of the truck platoon, including truck configuration, number of trucks, headway, and travel speed.

**Rationale:** The goal of the program is to determine the forces exerted by a given platoon on a bridge, they can come in many different forms, so flexi-

bility in the characteristics of the platoon are necessary for the relevance of the simulation.

**Fit Criterion:** At least one method of input(text based, dropdown list, etc) exists that allows users to specify those 4 characteristics of the platoon.

**FR.3** The Program should allow the user to visualize the effects of their truck platoon characteristic definitions on the final platoon.

**Rationale:** This is mainly to help the user verify that the inputs they put into the program correspond to the platoon they had in mind. As we are making a GUI, visual feedback is paramount to functionality.

**Fit Criterion:** There exists some visual representation of the truck platoon that changes to reflect the impact of changes in user input.

**FR.4** The Program should allow the user to define the characteristics of the bridge, including what type of bridge it is and its length.

**Rationale:** Different bridges will react to the same truck platoon differently, therefore specifying the relevant characteristics of the bridge is necessary for the relevance of the simulation.

**Fit Criterion** At least one method of input(text based, dropdown list, etc) exists that allows users to specify those 2 characteristics of the bridge.

**FR.5** The Program should allow the user to visualize the effects of their bridge characteristic definitions on the final bridge.

**Rationale:** This is mainly to help the user verify that the inputs they put into the program correspond to the bridge they had in mind. As we are making a GUI, visual feedback is paramount to functionality.

**Fit Criterion:** There exists some visual representation of the bridge that changes to reflect the impact of changes in user input.

**FR.6** The Program should allow the user to define which of the two solvers they are interested in using.

**Rationale:** as the MATLAB backend can solve for both the demand on a concerned section as the platoon drives along, as well as for which section has the highest maximum demand over the course of the whole trip, and these

are very different pieces of info, allowing the user to determine which they are currently interested in is important.

**Fit Criterion:** At least one method of input(text based, dropdown list, etc) exists that allows the user to choose which solver they are would like to use.

**FR.7** The Program should allow the user to define a section of concern on the bridge.

**Rationale:** The first solver revolves around calculating the demand on a certain point along the bridge as the truck platoon drives over, specifying what point it is that we care about is necessary for this function.

**Fit Criterion:** At least one method of input(text based, dropdown list, etc) exists that allows the user to determine a section of concern on the bridge.

**FR.8** The Program should allow the user to define a discretization length for their bridge.

**Rationale:** The second solver finds which section has the maximum demand placed on it over the course of the platoon's trip. The discretization length determines how many sections the bridge is split up into, which is necessary for the functioning of the second solver.

**Fit Criterion:** At least one method of input(text based, dropdown list, etc) exists that allows the user to define a discretization length for their bridge.

**FR.9** The Program should allow the user to define which type of demand placed on the bridge they are interested in, between shear forces and positive/negative moment.

**Rationale:** There are a variety of different demands placed on the bridge as the platoon drives over, and the MATLAB backend contains calculations for all 3 of the above mentioned demands. Allowing the user to define which of the 3 they are interested in seeing is necessary for the functionality of the simulation.

**Fit Criterion:** At least one method of input(text based, dropdown list, etc) exists that allows the user to define which of the 3 demands they are interested in simulating.

**FR.10** The Program should be capable of visualizing the results of the concerned section calculation for the user.

**Rationale:** This is essentially the main purpose of the GUI. Displaying the results of the MATLAB backend calculations visually to the user. This is one of the two main calculations to be represented, so this functionality is very necessary.

**Fit Criterion:** There exists some visualization of the mathematical results of the concerned section calculation.

## 2.4 Phase in Plan

# 3 Non-functional Requirements

## 3.1 Look and Feel Requirements

**NFR.?** The graphics will be informative.

**Rationale:** The user should gain value out of the presence of graphics.

**Fit Criterion:** Civil engineers who use the program will understand the different graphic elements used to represent bridge parts.

**Traceability:** ?.

## 3.2 Usability and Humanity Requirements

**NFR.?** The program will be intuitive to use.

**Rationale:** The program should be easy to use for its intended audience.

**Fit Criterion:** Civil engineers can use the program to generate a bridge system analysis within 5 minutes of introduction.

**Traceability:** ?.

**NFR.?** Program will have a user manual and (user-based) documentation provided.

**Rationale:** New users should be provided with resources to quickly start using the program.

**Fit Criterion:** User-based documentation covering usage and features of the program will be written.

**Traceability:** ?.

**NFR.?** The product will appear correctly on different display resolutions.

**Rationale:** Users of the product may wish to use it on displays of different resolutions.

**Fit Criterion:** The program will be viewed and its appearance validated on displays of different resolutions.

**Traceability:** ?.

**NFR.?** The product will allow for the resizing of text.

**Rationale:** Users of the product may wish to increase text size to allow for easier reading of the text.

**Fit Criterion:** We will ensure that the text size within the program is resizable, and that the program still functions correctly when the text size is changed.

**Traceability:** ?.

**NFR.?** The program will be easy to install and run.

**Rationale:** The program cannot be difficult to install for the users of our product.

**Fit Criterion:** We will measure the time it takes to download, install, and run our program.

**Traceability:** ?.

**NFR.?** (Not sure if usability or look and feel) UI elements which provide similar functionality will have a similar look.

**Rationale:** Having UI elements with comparable functionality be visually consistent will help with usability.

**Fit Criterion:** We will classify UI elements into categories and ensure that all elements within each categories are visually similar.

**Traceability:** ?.

### 3.3 Performance Requirements

**NFR.?** The program will safely handle unusual user inputs.

**Rationale:** Program should be robust and not prone to failure due to common misinputs.

**Fit Criterion:** The program will not freeze or crash as a direct result of a user providing inputs to the system.

**Traceability:** ?.

**NFR.?** The program will be able to handle missing dependencies.

**Rationale:** The program should be able to handle and warn of absent files.

**Fit Criterion:** The program will produce an error message when MATLAB scripts are absent or unable to run.

**Traceability:** ?.

**NFR.?** UI elements will react promptly to user input.

**Rationale:** The users of the program will want the UI to react quickly to their input.

**Fit Criterion:** The UI will graphically update to indicate it has acknowledged user inputs within 100ms of the input.

**Traceability:** ?.

**NFR.?** UI will not be unreasonably slow.

**Rationale:** UI should not introduce substantial delay beyond what is needed to calculate results.

**Fit Criterion:** The program delay when calculating and displaying results will not exceed the underlying MATLAB script's execution time by 10%.

**Traceability:** ?.

**NFR.?** The program will be precise.

**Rationale:** The program must be reasonably precise to provide value for



simulating bridges under load.

**Fit Criterion:** Calculations are accurate to within 1% relative error of similar bridge simulation engines.

**Traceability:** ?.

### 3.4 Operational and Environmental Requirements

**NFR.?** The program will run without slowdown on expected users' (MTO engineers) computers.

**Rationale:** The program must be able to run within requirements on the computers that it is intended to be used on.

**Fit Criterion:** Performance testing will be done on a computer with the same (or reasonably similar) hardware.

**Traceability:** ?.

### 3.5 Maintainability and Support Requirements

**NFR.?** The product shall be easily maintainable.

**Rationale:** The code must be easily maintainable to allow for future bug fixes and/or feature additions.

**Fit Criterion:** We will use file length, method length, and nesting depth as our primary indicators of code maintainability.

**Traceability:** ?.

### 3.6 Security Requirements

N/A; this project does not involve significant communication of user data.

### 3.7 Cultural Requirements

**NFR.?** The program should be able to be easily translated into other languages.

**Rationale:** People who don't speak English may wish to use the program, especially French speakers as Canada is a bilingual country.

**Fit Criterion:** Localization process to support French will not take over one week to complete.

**Traceability:** ?.

### 3.8 Legal Requirements

**NFR.?** Private MTOBridge assets will not be exposed for easy access to users.

**Rationale:** The client has expressed that their assets should be held confidential.

**Fit Criterion:** Received assets including MATLAB scripts will be excluded or compiled when in public repositories and distributed.

**Traceability:** ?.

### 3.9 Health and Safety Requirements

N/A; this project is only concerned with the graphical representation of the underlying bridge calculations.

## 4 Project Issues

### 4.1 Open Issues

N/A; there are currently no open issues for our project.

## **4.2 Off-the-Shelf Solutions**

N/A; since we are using novel calculations designed specifically for this project, there are no comparable off-the-shelf solutions to our product.

## **4.3 New Problems**

N/A; there are currently no new problems for our project

## **4.4 Tasks**

N/A; there are currently no required tasks for our project.

## **4.5 Migration to the New Product**

N/A; this product is novel and would not be replacing an existing product.

## **4.6 Risks**

## **4.7 Costs**

No monetary costs are involved in creating this product. The only costs in developing the product are the project team's time, and Dr. Yang's and their graduate students' time.

## **4.8 User Documentation and Training**

An extensive user manual with case study examples will be produced alongside the product. This documentation will allow for bridge engineers to become familiar and comfortable using the software. We will also train Dr. Cancan Yang and their graduate students to use the product so that they can effectively present the product to the MTO.

## **4.9 Waiting Room**

N/A; we don't currently have any requirements in the project waiting room.

## 4.10 Ideas for Solutions

N/A; we have not currently made any decisions about solution ideas.

## References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.

## 5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

### 5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC\_CONSTANTS. Their values are defined in this section for easy maintenance.

### 5.2 Likely Changes

### 5.3 Traceability Matrix

### 5.4 Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.
2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?