

Module Interface Specification for Truss Tool

Maryam Valian

March 30, 2023

1 Revision History

Date	Version	Notes
16/03/2023	1.0	Initial Draft
17/03/2023	1.1	Update Control module
19/03/2023	1.2	Update formulas
20/03/2023	1.3	Update other modules
23/03/2023	1.4	Update based on first Reviewer Feedback
30/03/2023	1.5	Update based on second Reviewer Feedback

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at [here](#).

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	1
6	MIS of Control module	3
6.1	Module	3
6.2	Uses	3
6.3	Syntax	3
6.3.1	Exported Constants	3
6.3.2	Exported Access Programs	3
6.4	Semantics	3
6.4.1	State Variables	3
6.4.2	Environment Variables	4
6.4.3	Assumptions	4
6.4.4	Access Routine Semantics	4
6.4.5	Local Functions	5
7	MIS of Input Module	5
7.1	Module	5
7.2	Uses	5
7.3	Syntax	5
7.3.1	Exported Constants	5
7.3.2	Exported Access Programs	5
7.4	Semantics	5
7.4.1	State Variables	5
7.4.2	Environment Variables	6
7.4.3	Assumptions	6
7.4.4	Access Routine Semantics	6
8	MIS of Specification Parameters	7
8.1	Module	7
8.2	Uses	7
8.3	Syntax	7
8.3.1	Exported Constants	7
8.3.2	Exported Access Programs	8
8.4	Semantics	8

8.4.1	State Variables	8
8.4.2	Environment Variables	8
8.5	MIS of Support Reaction Module	8
8.6	Module	8
8.7	Uses	8
8.8	Syntax	8
8.8.1	Exported Constants	8
8.8.2	Exported Access Programs	9
8.9	Semantics	9
8.9.1	State Variables	9
8.9.2	Environment Variables	9
8.9.3	Assumptions	9
8.9.4	Access Routine Semantics	9
9	MIS of Internal Forces Module	9
9.1	Module	9
9.2	Uses	10
9.3	Syntax	10
9.3.1	Exported Constants	10
9.3.2	Exported Access Programs	10
9.4	Semantics	10
9.4.1	State Variables	10
9.4.2	Environment Variables	10
9.4.3	Assumptions	10
9.4.4	Access Routine Semantics	10
9.5	MIS of Output Verification Module	11
9.6	Module	11
9.7	Uses	11
9.8	Syntax	11
9.8.1	Exported Constants	11
9.8.2	Exported Access Programs	11
9.9	Semantics	11
9.9.1	State Variables	11
9.9.2	Environment Variables	11
9.9.3	Assumptions	11
9.9.4	Access Routine Semantics	11
9.9.5	Assumptions	12
9.10	MIS of Output Module	12
9.11	Module	12
9.12	Uses	12
9.13	Syntax	12
9.13.1	Exported Constants	12
9.13.2	Exported Access Programs	12

9.14	Semantics	12
9.14.1	State Variables	12
9.14.2	Environment Variables	12
9.14.3	Assumptions	12
9.15	12
9.15.1	Access Routine Semantics	12
10	Appendix	14

3 Introduction

The following document details the Module Interface Specifications for Truss Tool is software designed for engineers and students to analyze a truss.

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at [Truss Tool repository](#).

4 Notation

The structure of the MIS for modules comes from [Hoffman and Strooper \(1995\)](#), with the addition that template modules have been adapted from [Ghezzi et al. \(2003\)](#). The mathematical notation comes from Chapter 3 of [Hoffman and Strooper \(1995\)](#). For instance, the symbol $:=$ is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$. Capital letters are used to indicate sequenced data type.

The following table summarizes the primitive data types used by Truss Tool.

Data Type	Notation	Description
character	char	a single symbol or digit
integer	\mathbb{Z}	a number without a fractional component in $(-\infty, \infty)$
natural number	\mathbb{N}	a number without a fractional component in $[1, \infty)$
real	\mathbb{R}	any number in $(-\infty, \infty)$

The specification of Truss Tool uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, Truss Tool uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	Level 2
Hardware-Hiding Module	
Behaviour-Hiding Module	Input parameters module Input verification module Specification parameters module Output format module Output verification module Support reactions module. Force decomposing module Internal force module Control module
Software Decision Module	Sequence data structure module linear equation solver module

Table 1: Module Hierarchy

6 MIS of Control module

6.1 Module

Main

6.2 Uses

Input Module, Reaction Module, Internal Forces Module, Output Verification Module

6.3 Syntax

6.3.1 Exported Constants

Not applicable.

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
Main	-	-	-

6.4 Semantics

6.4.1 State Variables

from R1 and R2:

$n : \mathbb{N}$

$m : \mathbb{N}$

$J_n : \mathbb{R}$

$M_m : \mathbb{N}$

$F_m : \mathbb{R}$

$sp : \mathbb{N}$

$sr : \mathbb{N}$

from R3:

$px : \mathbb{R}$

$py : \mathbb{R}$

$ry : \mathbb{R}$

from R4:

$I_m : \mathbb{R}$

You don't have state vars
for main. You have
put your input in the
Input module. You don't
want the same state info.
in two different places.

6.4.2 Environment Variables

This module has external interaction with an input file, an output file.

6.4.3 Assumptions

The path and the name of the given input file are correct.

6.4.4 Access Routine Semantics

Main():

- transition: Modifies the state of the Input module and the environment variables for the Output modules by following these steps:

Get (filename: String) from user which indicates input file. LoadParams(filename).

#Find Support reactions (P_x, P_y, R_y)

$$P_x = -\sum F_x$$

This should be in your
other module

$$A = \begin{bmatrix} 1 & 1 \\ J_x(sp) & J_x(sr) \end{bmatrix}$$
$$B = \begin{bmatrix} -\sum F_y(i) \\ -\sum (J_x(i) * F_y(i)) \end{bmatrix}$$

$$(P_y, R_y) = \text{Solve}(A, B)$$

#Find Internal Forces I_m

$$\theta = \text{Decomp}(n, m, J, M)$$

#Compute a_{ij}, b_j elements of A, B from equilibrium equations:

$$a_{ij} = I_m * \cos \theta_m, b_j = \sum_x \text{All forces and reactions at } j$$

or

$$a_{ij} = I_m * \sin \theta_m, b_j = \sum_y \text{All forces and reactions at } j$$

$$I = \text{Solve}(A, B)$$

#Output P_x, P_y, R_y, I

verify-Output(P_x, P_y, R_y, I)

output(file-name, P_x, P_y, R_y, I)

- output: output file
- exception: Input file not found

✓ If it does, then you should
name these enviro
variables

You have
access programs
from your module
to do these
tasks. You should
show the calls
to those access
programs here,
rather than give
a pseudo code
overview.

6.4.5 Local Functions

None.

7 MIS of Input Module

7.1 Module

Inputs

7.2 Uses

Input Verification Module

7.3 Syntax

Name	In	Out	Exceptions
load_params	string	-	Input File Error
verify_params	-	-	Input Parameters Error
n	-	\mathbb{N}	
m	-	\mathbb{N}	
J_n	-	\mathbb{R}	
M_m	-	\mathbb{N}	
F_m	-	\mathbb{R}	
sp	-	\mathbb{N}	
sr	-	\mathbb{N}	

7.3.1 Exported Constants

Not applicable.

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
Inputs	input plain text file	Parameters	FileError

7.4 Semantics

7.4.1 State Variables

from R1 and R2:

$n : \mathbb{N}$

$m : \mathbb{N}$

add comments to ~~formal~~
to define these variables
what are the types of given
variables?

$J_n : \mathbb{R}$
 $M_m : \mathbb{N}$
 $F_m : \mathbb{R}$
 $sp : \mathbb{N}$
 $sr : \mathbb{N}$

does this make sense? what is J_n ? It looks
 like a function with the
 argument x ?

7.4.2 Environment Variables

InputFile: sequence of strings in the text file.

7.4.3 Assumptions

- Load parameters will be called before the values of any state variables will be accessed.
- The file contains the string equivalents of the numeric values for each input parameter. The order is important.

7.4.4 Access Routine Semantics

Load-Parameter():

- transition: Modifies the state variables from the input file:
 - Read sequentially from file and loads state variables from R1 and R2.
 - verify-input()
- output: None
- exception: file-Error, input-type-mismatch-Error

Verify-Input():

- transition: Modifies the state variables from the input file:
 - Read constants from specification module.
 - Check constraints given in Table 2.
- output: None
- exception: Bad-parameters-Error

Table 2: Constraint check table

Constraint	Error
$n_{min} \leq n \leq n_{max}$	BadInputSize
$m_{min} \leq m \leq m_{max}$	BadInputSize
$m < n$	BadInputSize
$ J \neq n$	InconsistentInput
$ M \neq m$	InconsistentInput
$ F \neq n$	InconsistentInput
$J_{min} \leq J_j \leq J_{max}$	BadInputSize
$F_{min} \leq F_j \leq F_{max}$	BadInputSize
$M_{j,0} = M_{j,1}$	BadMember
$Sp > n, Sp < 0$	BadSupport
$Sr > n, Sr < 0$	BadSupport
$Sp = Sr = 0$	NoSupportDefined

what is J rdn?
 J is a set?
 J is a sequence?
 Use J_j for sequence.

8 MIS of Specification Parameters

8.1 Module

Specification

8.2 Uses

None.

8.3 Syntax

8.3.1 Exported Constants

From Table.2 SRS:

$n_{min} := 3$
 $n_{max} := 20$
 $m_{min} := 3$
 $m_{max} := 30$
 $J_{max}(i, j) := 200$
 $J_{min}(i, j) := 0$
 $F_{max} := 35000$

what are i & j ?

$F_{min} := -35000$

8.3.2 Exported Access Programs

8.4 Semantics

8.4.1 State Variables

None.

8.4.2 Environment Variables

None.

8.5 MIS of Support Reaction Module

8.6 Module

React

8.7 Uses

Sequence data structure module.
linear equation solver module.

8.8 Syntax

Name	In	Out	Exceptions
React	params	reacts	ReactionFailed Error
n	\mathbb{N}	-	
F_j	\mathbb{R}	-	
J_j	\mathbb{R}	-	
Sp	\mathbb{N}	-	
Sr	\mathbb{N}	-	
P_x	-	\mathbb{R}	
P_y	-	\mathbb{R}	
R_y	-	\mathbb{R}	

8.8.1 Exported Constants

None.

8.8.2 Exported Access Programs

Name	In	Out	Exceptions
React	$(J, F$ $arrayofreal, sp, sr$ $Integer)$	$: (P_x, P_y, R_y : Real)$ $:$	ReactFailed

8.9 Semantics

8.9.1 State Variables

None.

8.9.2 Environment Variables

None.

8.9.3 Assumptions

None.

8.9.4 Access Routine Semantics

- Output:

$$Out := P_x = - \sum F_x$$

$$A = \begin{bmatrix} 1 & 1 \\ J_x(sp) & J_x(sr) \end{bmatrix}$$

$$B = \begin{bmatrix} - \sum F_y(i) \\ - \sum (J_x(i) * F_y(i)) \end{bmatrix}$$

$$Out := (P_y, R_y) = Solve(A, B)$$

9 MIS of Internal Forces Module

9.1 Module

Internal

9.2 Uses

Force decomposing module
Sequence data structure module
linear equation solver module

9.3 Syntax

9.3.1 Exported Constants

None.

9.3.2 Exported Access Programs

Name	In	Out	Exceptions
Decomp	$\theta(M_m) : \mathbb{R}^2 \rightarrow \mathbb{R}$	Θ_m	decompFailed
Internal	$I_m(F_j, \theta_m) : \mathbb{R} * \mathbb{R}^2 \rightarrow \mathbb{R}$	I_m	InternalFailed

Exception decompFailed will show to the user if a correct member is not defined by the user so the angle between the member and the x-axis cannot be calculated. Exception InternalFailed will show to the user if the equations cannot be solved.

9.4 Semantics

9.4.1 State Variables

None.

9.4.2 Environment Variables

None.

9.4.3 Assumptions

None.

9.4.4 Access Routine Semantics

- Output: $Out := I$ where:

$$a_{ij} = I_m * \cos \theta_m, b_j = \sum_x \text{All forces and reactions at } j$$

or

There are too many
undefined symbols here.

Is this decomp or
Internal?

One of these functions
is not defined in
the spec.

$$a_{ij} = I_m * \sin \theta_m, b_j = \sum_y \text{All forces and reactions at } j$$

$$I = \text{Solve}(A, B)$$

9.5 MIS of Output Verification Module

9.6 Module

verify-out

9.7 Uses

None.

9.8 Syntax

9.8.1 Exported Constants

None.

9.8.2 Exported Access Programs

Name	In	Out	Exceptions
Verify-Out	$check(I_m, J_j) : \mathbb{R} * \mathbb{R} \rightarrow Boolean$	Valid/NotValid	verifyFailed

9.9 Semantics

9.9.1 State Variables

None.

9.9.2 Environment Variables

None.

9.9.3 Assumptions

None.

9.9.4 Access Routine Semantics

- Exception: $\text{Exep} := (\sum (I_m(n) + F_m(n), P_x(n), P_y(n), R_y(n)) \neq 0) \Rightarrow \text{VerifyFailed}$

9.9.5 Assumptions

None.

9.10 MIS of Output Module

9.11 Module

Output

9.12 Uses

None.

9.13 Syntax

9.13.1 Exported Constants

output file name.

9.13.2 Exported Access Programs

None.

9.14 Semantics

9.14.1 State Variables

None.

9.14.2 Environment Variables

output file

9.14.3 Assumptions

None.

9.15

Assumptions None.

9.15.1 Access Routine Semantics

- transition: Write to file the following: the input parameters from Param, and the calculated values I_m, P_x, P_y, R_y .

References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL <http://citeseer.ist.psu.edu/428727.html>.

there is some confusion on how to write a mathematical spec. Focus on the type of all symbols and having everything defined within the scope where it is mentioned. I think the design is still taking shape at this point in the development. Once the code is working, you can revise the MS to match the code.

10 Appendix

Not applicable.