Module Interface Specification for Solar Water Heating Systems Incorporating Phase Change Material

Maya Grab and Brooks MacLachlan

July 29, 2016

Contents

1 Introduction

The following document details the Module Interface Specifications for the implemented modules in a program simulation Solar Water Heating System with Phase Change Material. It is intended to ease navigation through the program for design and maintenance purposes. Complementary documents include the System Requirement Specifications and Module Guide.

2 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 Format Module Input Parameters Module Input Verification Module Output Format Module Output Verification Module Temperature ODEs Module Energy Equations Module Control Module
Software Decision Module	Sequence Data Structure Module ODE Solver Module Plotting Module

Table 1: Module Hierarchy

3 MIS of Control Module

3.1 Module Name: main.m

3.2 Uses

3.2.1 Imported Data Types

Uses Input Parameters Module **Imports** params := structure

3.2.2 Imported Access Programs

Uses Input Format Module **Imports** load_params
Uses Energy Module **Imports** energy1, energy2, energy3
Uses Temperature Module **Imports** temperature1, temperature2, temperature3
Uses Event Module **Imports** event1, event2
Uses Plot Module **Imports** plots

3.3 Interface Syntax

3.3.1 Exported Access Programs

Name	In	Out	Exceptions
main	String (file)	Modifies the screen environment Modifies the output file to contain its state variables	Various

3.4 Interface Semantics

3.4.1 State Variables

t : column vector
T : column vector
Ew : column vector
Ep : column vector
Etot : column vector

3.4.2 Assumption

[Should this be commented out if there is no content? —BM]

3.4.3 Invariant

None

3.4.4 Access Program Semantics

Input:
Main.m will accept a valid file name string which is accessible to the current Matlab path. The data in the provided file must comply with the format specifications required by the Input Format Module.

Exceptions:
Potential exceptions are invalid file names or paths. Other exceptions are possible within the Input Format Module and Input Verification Module due to inappropriate input.

Output:
Main.m will print to the screen messages regarding phase changes of the PCM.
Main.m will request the Output Format Module to produce a file with the same file name as the input string, with the extension .out. The file will contain the state variables of main at the end of the simulation.
Main.m will request the Plotting Module to produce energy and temperature graphs from its state variables.

4 MIS of Input Parameters Module

4.1 Module Name: ParamT

4.2 Interface Syntax

4.2.1 Exported Data Types

params := structure

4.3 Interface Semantics

Params is a data structure designed to store the input information entered by the Input Format module.

5 MIS of Input Format Module

5.1 Module Name: load_params

5.1.1 Imported Data Types

Uses Input Parameters Module **Imports** params := structure

5.2 Interface Syntax

5.2.1 Exported Access Programs

Name	In	Out	Exceptions
load_params	String (file)	Struct	Inappropriately formatted input file

5.3 Interface Semantics

5.3.1 Access Program Semantics

Input:
load_params will accept a valid file name string which is accessible to the current Matlab path. The readable data in the given file must be numeric only, with all comments appropriately signaled.

Exceptions:
Potential exceptions may occur when the given file containing input data is inappropriately formatted.

Output:
When given valid data load_params will return a structure containing all inputted data under appropriate field names, in addition to some relevant calculated data fields.

6 MIS of Input Verification Module

6.1 Module Name: verify_params

6.1.1 Imported Data Types

Uses Input Parameters Module **Imports** params := structure

6.2 Interface Syntax

6.2.1 Exported Access Programs

Name	In	Out	Exceptions
verify_params	Struct	none	Various (see appendix)

6.3 Interface Semantics

6.3.1 Access Program Semantics

Input:
verify_params will accept a data structure containing the fields created by the Input Format Module.

Exceptions:
Data which does not comply with the data constraints specifications detailed in the SRS document for this project will yield one of the potential exceptions or warnings as listed in the appendix of this document.

Output:
None

7 MIS of Temperature Modules

7.1 Module Name: temperature1, temperature2, temperature3

7.2 Interface Syntax

7.2.1 Exported Access Programs

Name	In	Out	Exceptions
temperature1	real vector struct	vector	Vector dimensions Data structure missing fields
temperature2	real vector struct	vector	Vector dimensions Data structure missing fields
temperature3	real vector struct	vector	Vector dimensions Data structure missing fields

7.3 Interface Semantics

7.3.1 Assumption

The Control Module and the Event Module handled the different cases correctly, ensuring that each temperature function necessarily receives valid numerical input for the case it is meant to handle.

7.3.2 Access Program Semantics

Input:
The Temperature Module functions require a time value, a vector containing at least two values, and a data structure containing the fields created by the Input Format Module.

Exceptions:
Potential exceptions may occur when given inappropriately sized vectors or a data structure lacking the required fields.

Output:
The Temperature Module will return a vector containing numerical solutions to the differential equations it governs.

7.3.3 Considerations

Note: the three exported access functions are implemented in three separate function files.

8 MIS of Energy Modules

8.1 Module Name: energy1, energy2, energy3

8.2 Interface Syntax

8.2.1 Exported Access Programs

Name	In	Out	Exceptions
energy1	matrix struct	vector	Matrix dimensions Data structure missing fields
energy2	matrix struct	vector	Matrix dimensions Data structure missing fields
energy3	matrix struct	vector	Matrix dimensions Data structure missing fields

8.3 Interface Semantics

8.3.1 Assumption

The Control Module and the Event Module handled the different cases correctly, ensuring that each energy function necessarily receives valid numerical input for the case it is meant to handle.

8.3.2 Access Program Semantics

Input:
The Energy Module functions require a matrix containing at least three columns of numerical data, and a data structure containing the fields created by the Input Format Module.

Exceptions:
Potential exceptions may occur when given an inappropriately sized matrix or a data structure lacking the required fields.

Output:
The Energy Module will return vectors containing numerical solutions to the differential equations it governs.

8.3.3 Considerations

Note: the three exported access functions are implemented in three separate function files.

9 MIS of Event Modules

9.1 Module Name: event1, event2

9.2 Interface Syntax

9.2.1 Exported Access Programs

Name	In	Out	Exceptions
event1	real vector struct	real int int	None
event2	real vector struct	real int int	Vector dimensions Data structure missing fields

9.3 Interface Semantics

9.3.1 Access Program Semantics

Input:
The Event Module functions require a numerical value, a vector containing at least three numerical values, and a data structure containing the fields created by the Input Format Module.

Exceptions:
Potential exceptions may occur when given inappropriately sized vectors or a data structure lacking the required fields.

Output:
The Event Module will return three numerical values.

9.3.2 Local Constants

direction = 0
isterminal = 1

9.3.3 Considerations

Note: the two exported access functions are implemented in two separate function files.

10 MIS of Plotting Module

10.1 Module Name: plots

10.2 Uses

10.3 Interface Syntax

10.3.1 Exported Access Programs

Name	In	Out	Exceptions
plots	vector matrix vector vector	displays figures	Matrix dimensions Vector dimensions

10.4 Interface Semantics

10.4.1 Assumption

All vectors and matrices are handled correctly by the Control Module to be the correct size.

10.4.2 Access Program Semantics

Input:
The Plotting Module takes a matrix containing at least two columns of numerical data, and three different vectors containing numerical data.

Exceptions:
Potential exceptions may occur when given inappropriately sized matrices and vectors.

Output:
The Plotting Module produces figures of the plotted data it has received as input.
[Is it okay that the Exceptions contradict the Assumption? —BM]

11 MIS of Output Verification Module

11.1 Module Name: verify_output

11.1.1 Imported Data Types

Uses Input Parameters Module **Imports** params := structure

11.2 Interface Syntax

11.2.1 Exported Access Programs

Name	In	Out	Exceptions
	struct		
	vector		
	matrix		Matrix dimensions
	vector		Vector dimensions
verify_output	vector	none	Data structure missing fields

11.3 Interface Semantics

11.3.1 Assumption

All vectors and matrices are handled correctly by the Control Module to be the correct size.

11.3.2 Access Program Semantics

Input:
verify_output will accept a data structure containing the fields created by the Input Format Module, three vectors containing numerical data, and a matrix containing at least two columns of numerical data.

Exceptions:
Potential exceptions may occur when given inappropriately sized matrices and vectors or a data structure lacking the required fields.

Output:
None

12 MIS of Output Format Module

12.1 Module Name: output

12.2 Interface Syntax

12.2.1 Exported Access Programs

Name	In	Out	Exceptions
	string (file)		
	vector		
	matrix		
	vector		
	vector		Matrix dimensions
	vector		Vector dimensions
output	struct	prints to file	Data structure missing fields

12.3 Interface Semantics

12.3.1 Assumption

All vectors and matrices are handled correctly by the Control Module to be the correct size.

12.3.2 Access Program Semantics

Input:
The Output Format Module takes a string, a matrix containing at least two columns of numerical data, four different vectors containing numerical data, and a data structure as produced by the Input Format Module.

Exceptions:
Potential exceptions may occur when given inappropriately sized matrices and vectors or a data structure lacking the required fields.

Output:
The Output Format Module produces a file with the given string as a name, which contains a formatted list of data given by the input arrays and data structure.

13 Appendix

Table 2: Standard Input Variables

Var	**Typical Value**
L	1.5 m
D	0.412 m
V_P	0.05 m^3
A_P	1.2 m^2
ρ_P	1007 kg/m^3
T^P_{melt}	44.2 °C
C^S_P	1760 J/(kg °C)
C^L_P	2270 J/(kg °C)
H_f	211600 J/kg
A_C	0.12 m^2
T_C	50 °C
ρ_W	1000 kg/m^3
C_W	4186 J/(kg °C)
h_C	1000 W/(m^2 °C)
h_P	1000 W/(m^2 °C)
T_{init}	40 °C
t_{final}	50000 s
AbsTol	10^{-10}
RelTol	10^{-10}
ConsTol	10^{-3}

Table 3: Possible Exceptions

Message ID	**Error Message**
input:L	error: Tank length must be > 0
input:diam	error: Tank diameter must be > 0
input:Vp	error: PCM volume must be > 0
input:VpVt	error: PCM volume must be $<$ tank volume
input:Ap	error: PCM area must be > 0
input:rho_p	error: rho_p must be > 0
input:Tmelt	error: Tmelt must be > 0 and $< T_C$
input:C_ps	error: C_ps must be > 0
input:C_pl	error: C_pl must be > 0
input:Hf	error: Hf must be > 0
input:Ac	error: Ac must be > 0
input:Tc	error: Tc must be > 0 and < 100
input:rho_w	error: rho_w must be > 0
input:C_w	error: C_w must be > 0
input:hc	error: hc must be > 0
input:hp	error: hp must be > 0
input:Tinit	error: Tinit must be > 0 and < 100
input:TcTinit	error: Tc must be $>$ Tinit
input:TinitTmelt	error: Tinit must be $<$ Tmelt
input:tfinal	error: tfinal must be > 0